Understanding and Improving Knowledge Distillation for Quantization-Aware Training of Large Transformer Encoders

Minsoo Kim[1], Sihwa Lee[2], Sukjin Hong[3], Du-Seong Chang[3], and Jungwook Choi[1,2]*

[1]Department of Electronic Engineering, Hanyang University
[2]Department of Artificial Intelligence, Hanyang University
Seoul, Republic of Korea
{minsoo2333, macto94, choij}@hanyang.ac.kr,
[3]KT, Seoul, Republic of Korea
{sukjin.hong, dschang}@kt.com

Abstract

Knowledge distillation (KD) has been a ubiquitous method for model compression to strengthen the capability of a lightweight model with the transferred knowledge from the teacher. In particular, KD has been employed in quantization-aware training (QAT) of Transformer encoders like BERT to improve the accuracy of the student model with the reduced-precision weight parameters. However, little is understood about which of the various KD approaches best fits the QAT of Transformers. In this work, we provide an in-depth analysis of the mechanism of KD on attention recovery of quantized large Transformers. In particular, we reveal that the previously adopted MSE loss on the attention score is insufficient for recovering the self-attention information. Therefore, we propose two KD methods; attention-map and attention-output losses. Furthermore, we explore the unification of both losses to address task-dependent preference between attention-map and output losses. The experimental results on various Transformer encoder models demonstrate that the proposed KD methods achieve state-of-the-art accuracy for QAT with sub-2-bit weight quantization.

1 Introduction

Knowledge distillation (KD) (Hinton et al., 2015) is a transfer learning framework to pass on knowledge of a large model (the teacher) to a lightweight model (the student). Numerous KD methods have been developed regarding the source of knowledge and distillation objectives. In many cases, Kullback-Leibler divergence (KL-Div) is used as a default distillation objective to match the soft labels of the teacher and the student (Hinton et al., 2015; Sanh et al., 2019). But further studies on KD suggest that internal representations also convey the intermediate knowledge of the teacher (Sun et al., 2019; Aguilar et al., 2020). Thus minimizing the distance (e.g., mean squared error, MSE)

---
*Corresponding Author

of hidden state knowledge (HSK) of layers between the teacher and the student has also been proposed (Sun et al., 2019; Liu et al., 2021a). KD becomes an essential model compression technique for efficiently deploying large-scale Transformer-based language models. For example, a popular Transformer encoder model, BERT, contains hundreds of millions of parameters, incurring profound memory and computation overhead (Devlin et al., 2019). These large-scale models require extreme compression to reduce the model footprint by 10 to 100 times. Therefore, extensive studies have been accomplished to distill efficient student models (Sanh et al., 2019; Sun et al., 2019; Jiao et al., 2020; Wang et al., 2020, 2021), but their focus is limited to achieving fewer parameters.

Quantization-aware training (QAT) stands out for its recent success in reducing not only the memory requirements but also the computational complexity of Transformer models (Bhandare et al., 2019; Zafrir et al., 2019; Kim et al., 2021a). Although QAT reflects quantization errors during the forward pass computation of stochastic gradient descent to train a model more robust to quantization errors, quantizing weight parameters of Transformers to a precision lower than 2-bits degrades the accuracy. Therefore, many recent QAT techniques employed the KD framework to distill the capability of the full-precision teacher to the student model with reduced-precision parameters (Zhang et al., 2020; Bai et al., 2021; Jin et al., 2021; Li et al., 2022). However, little is understood about which of the various KD approaches best fits the QAT of large Transformers. Without careful justification, most prior works adopted the layer-wise distillation of the attention score and the Transformer output with the MSE loss in addition to the basic KL-Div loss on the model output. Therefore, it is unclear if such KD setting is the most helpful for QAT on large-scale Transformer encoders like BERT-Large.

In this work, we provide an in-depth analysis of KD on attention recovery for QAT of Transformers in terms of the knowledge sources and the objectives. We first reveal that all-layer KD of the intermediate Transformer layer is essential for QAT, in contrast to the KD-based model compression. In the case of BERT-Base, we further discover that the KL-Div-based KD on attention-map (called attention-map loss) outperforms the prior KD technique that takes MSE loss on the attention score. However, the attention-map loss is insufficient for the large Transformer encoders since weight quantization disrupts attention propagation for specific NLP tasks when there are many layers. Therefore, we devise an insightful KD, MSE loss on attention output (called attention-output loss), and help preserve attention recovery along with many layers. The proposed attention-map and output losses and their combination are evaluated on various Transformer encoder models (BERT-Base/Large and a BERT-like Korean language model (ULM)). The experimental results demonstrate that the proposed KD methods significantly boost the model accuracy surpassing the state-of-the-art for QAT with aggressive sub-2-bit weight quantization.

We summarize our contributions as follows:

- We improve the prior KD techniques for QAT to boost the accuracy of large Transformer encoders.

- We quantitatively reveal that the *attention-map loss* (based on KL-Div) outperforms the existing attention-score loss (based on MSE). The proposed attention-map loss is particularly beneficial for the BERT-Base model.

- We discover the task-dependent attention characteristics, particularly noticeable in BERT-Large. In particular, we reveal that specific tasks on large Transformers suffer homogenization of attention output when weights are quantized. We propose a new KD method, *attention-output loss*, to address this issue.

- We further explore the potential of *unifying the attention-map and output losses* to handle task-dependent attention characteristics ubiquitously.

- We evaluate the proposed KD methods on various large-scale Transformer encoders and NLP tasks, achieving state-of-the-art accuracy for sub-2-bit aggressive QAT.
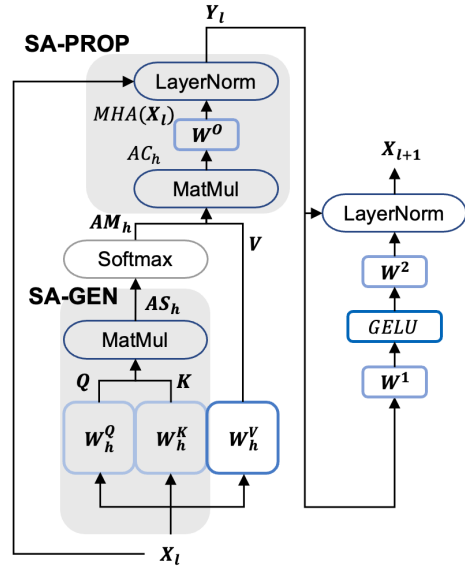

Figure 1: The Transformer layer architecture.

2 Related Work

2.1 Transformer Encoder Model

Transformer-based encoder models like BERT (Devlin et al., 2019) has been widely adopted for natural language processing (NLP) tasks such as question answering and language inference. As Fig. 1 shows, these models are built with Transformer layers consisting of Multi-Head Attention (MHA) and Feed-Forward Network (FFN) (Vaswani et al., 2017). The 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 N_H be the number of attention heads and $d_h = d/N_H$. $\mathbf{W}_h^Q, \mathbf{W}_h^K, \mathbf{W}_h^V \in \mathbb{R}^{d \times d_h}$ 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 ($\mathbf{AS}_h = \mathbf{QK}^\top$), attention map ($\mathbf{AM}_h = \mathrm{Softmax}_h(\frac{\mathbf{AS}_h}{\sqrt{d}})$), and attention context ($\mathbf{AC}_h = \mathbf{AM}_h\mathbf{V}$). MHA is defined as:

$$\mathrm{MHA}(\mathbf{X}_l) = \mathrm{Concat}(\mathbf{AC}_1, ...\mathbf{AC}_{N_H})\mathbf{W}^O. \quad (1)$$

Motivated by (Kobayashi et al., 2020), MHA can be re-written per each token i:

$$\mathrm{MHA}(\mathbf{X}_l)(i) = \sum_{j=1}^{n} \alpha_{i,j} f(\mathbf{X}_l(j)), \quad (2)$$

where $f(x) := (x\mathbf{W}^V + \mathbf{b}^V)\mathbf{W}^O$ and $\alpha_{i,j}$ is j'th attention probability of i'th token in $\mathbf{AM}_h$. Therefore, MHA can be decomposed into two parts: self-attention generation (SA-GEN) corresponding to

the attention map (α), and self-attention propagation (SA-PROP) corresponding to $f(x)$. Fig. 1 shows which part is SA-GEN and SA-PROP respectively.

FFN consists of two fully-connected layers with weight parameters $\mathbf{W}^1$ and $\mathbf{W}^2$:

$$\text{FFN}(\mathbf{Y}_l) = \text{GELU}(\mathbf{Y}_l\mathbf{W}^1 + \mathbf{b}^1)\mathbf{W}^2 + \mathbf{b}^2. \quad (3)$$

Therefore, output of a Transformer layer $\mathbf{X}_{l+1}$ is defined as:

$$\begin{aligned} \mathbf{Y}_l &= \text{LayerNorm}(\mathbf{X}_l + \text{MHA}(\mathbf{X}_l)), \\ \mathbf{X}_{l+1} &= \text{LayerNorm}(\mathbf{Y}_l + \text{FFN}(\mathbf{Y}_l)). \end{aligned} \quad (4)$$

Here, $\mathbf{Y}_l$ and $\mathbf{X}_{l+1}$ are called attention output (**AO**) and Transformer output, respectively.

2.2 Knowledge Distillation for Compression of Transformer Models

Knowledge distillation (KD) (Hinton et al., 2015) is a transfer learning framework that a lightweight model (the student) learns from the knowledge distilled from a cumbersome model (the teacher). Since KD provides the student information to reach the teacher's capability, KD has been widely adopted for model compression of large-scale Transformer models like BERT. A basic distillation approach is to match the probability distribution at the output of the teacher and student models via CE loss, as in DistilBERT (Sanh et al., 2019). In addition to this soft-label distillation, PKD (Sun et al., 2019) suggested KD on the normalized output of each Transformer layer, as distillation on the teacher's intermediate representations can benefit the student. MobileBERT (Sun et al., 2020) also employed per-head KD on the attention map and customized architecture for efficient Transformer computations. MiniLM and MiniLMv2 (Wang et al., 2020, 2021) further transferred relational knowledge from the self-attention map, but only at a single Transformer layer (located at the last or upper-middle). (Liu et al., 2021a) further claimed that distilling more intermediate representations does not necessarily help improve the accuracy of the student.

Although these KD-based compression techniques have developed efficient BERT structures, there has been limited understanding of KD on the model quantization. In particular, we are the first to quantitatively reveal that more distillation of the intermediate representations helps QAT reduce the accuracy gap between the quantized student and the full-precision teacher.

2.3 Quantization for BERT

Quantization is a promising technique to reduce the high inference cost of large-scale models without changing the model structure. Instead of representing numbers in 32-bit floating-point (FP32), employing fixed-point representation, such as 8-bit integer (INT8) quantization, has achieved significant speedup and storage savings for BERT (Zafrir et al., 2019; Kim et al., 2021a; Lin et al., 2021). However, direct quantization of weight parameters leads to degradation of the original model accuracy when 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 a precision lower than 2-bit. TernaryBERT (Zhang et al., 2020) represents each weight parameter into one of three values $\{-1, 0, 1\}$. TernaryBERT actively incorporates KD into QAT for improving accuracy degradation. To reduce the bit-precision, BinaryBERT (Bai et al., 2021) proposed a modified QAT procedure with a specific weight initialization for binary quantization. DQ-BART (Li et al., 2022) further combined model compression (via layer reduction) and quantization by exploiting KD.

Although KD has been a de-facto technique for QAT, there is a lack of understanding about why. In particular, the aforementioned QAT methods all employed the layer-wise KD on the self-attention score (AS_l) and Transformer output (X_l) along with the KD on soft labels. Considering numerous KD techniques with various choices for the knowledge sources and the objective, it is not clear if the current recipe helps QAT the most. This work investigates the prior layer-wise KD techniques and improves them with new objectives and knowledge sources.

3 Prior KD Techniques for QAT

In this section, we investigate prior KD techniques for QAT evaluated on BERT-Base. As discussed earlier, KD techniques commonly used for QAT include 1) all-layer distillation and 2) distillation on SA-GEN. First, we provide justification and improvement on these techniques. Then we further showcase the limitation when they are applied to large-scale Transformer encoders.
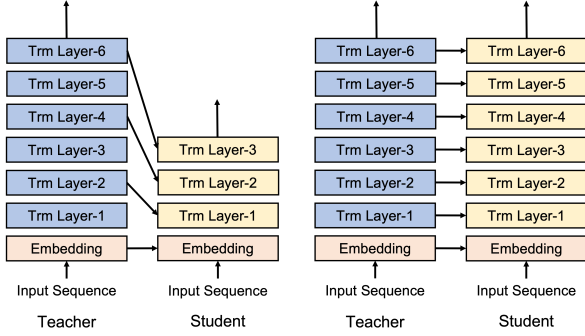
Figure 2: Illustration of layer selection strategy in model compression and model quantization. Left: Uniform mapping strategy. Right: All-layer distillation.
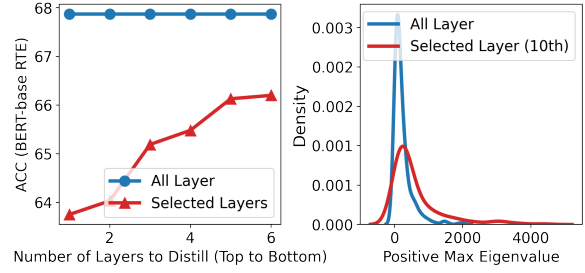

Figure 3: Comparison of (a) RTE accuracy and (b) Hessian max Eigenvalues Spectra between the selected-layer distillation and all-layer distillation in BERT-Base.

3.1 All-Layer Distillation for QAT

Generally, the internal representation of the teacher, such as a layer output, is widely used for knowledge distillation for model compression (Aguilar et al., 2020). However, there is a distinct difference in KD between typical model compression and quantization. For example, Fig. 2 shows two representative layer-to-layer mapping for KD: selected-layer distillation for model compression (left) and all-layer distillation for QAT (right). In the case of selected-layer distillation, the study showed that the marginal utilities of hidden state knowledge (HSK) diminish as more HSK has been distilled(Liu et al., 2021a). In contrast, most prior QAT methods applied KD on the Transformer output of all the layers. The structural equivalence of the teacher and the student of QAT methods makes this choice natural, but there is little justification.

We conjecture that quantization applied to the weight parameters disrupts the functionality of the Transformer layer, necessitating layer-wise guidance. To validate this conjecture, we conducted two experiments. First, we compared the accuracy of the uniformly selected layer distillation with a varying number of distilled layers. As shown in Fig. 3a, the accuracy grows along with the number of distilled layers, and all-layer distillation significantly outperforms selected-layer distillation. In addition, we compared the loss surface of the two distillation approaches after QAT in terms of Hessian max eigenvalues (Park and Kim, 2022). We adopt single layer selection distillation as a strategy of selected layer distillation. In particular, we used the method of selecting the 10th layer of BERT-Base model, which was most helpful in performance as a single layer selection distillation. As shown in Fig. 3b, all-layer distillation shows smaller magnitudes of

Eigenvalues, indicating a smoother loss surface. Therefore, we can conclude that the layer-wise distillation helps train the student with the quantized weight parameters.

3.2 Improve KD on Self-Attention Generation

We further investigate the objective of all-layer KD. As discussed earlier, prior QAT methods employed MSE loss on the attention score (called attention-score loss) for all-layer KD, as follows:

$$\mathcal{L}_{score} = \sum_{l=0}^{L-1} \text{MSE}(\mathbf{AS}_l^T, \mathbf{AS}_l^S). \quad (5)$$

Given that the attention map captures the correlation of one token to all the others, it is essential to maintain the relative importance of tokens. However, quantization in nature clamps and coarsely represents the weight parameters, making attention less distinguishable. We expect KD to help maintain disparity, but the attention-score loss is not a proper objective since it mainly focuses on logit matching (Kim et al., 2021b).

As an alternative, we propose to use the KL-Div loss on the attention-map (called attention-map loss) defined as follows:

$$\mathcal{L}_{map} = \frac{1}{N_h n} \sum_{h=1}^{N_h} \sum_{t=1}^{n} D_{KL}(\mathbf{AM}_{l,h,t}^T \| \mathbf{AM}_{l,h,t}^S). \quad (6)$$

Assuming that the temperature hyper-parameter (τ) is one, KL-Div focuses on label-matching (Kim et al., 2021b). Thus the relative importance of attention across tokens is better maintained with attention-map loss. Although the attention-map loss was previously employed in model compression (Wang et al., 2021), we are the first to quantitatively reveal the benefits of the attention-map loss in the context of QAT.
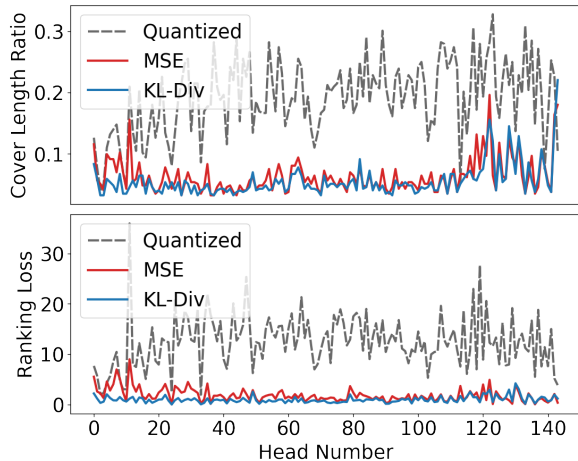
Figure 4: The cover length ratio and ranking loss per attention head in BERT-Base. X-axis: Number of attention heads. Quantized: Quantized model without applying QAT, KL-Div: attention-map loss, MSE: attention-score loss

We introduced two metrics that characterize the attention map to evaluate the proposed KD loss quantitatively. The cover length ratio captures the student's attention map deviation from the teacher's based on Top-K token coverage. The ranking loss (Liu et al., 2021b) shows the similarity in the attention rankings of the teacher and the student. Fig. 4 compares the cover length ratio and the ranking loss of every attention head of BERT-Base on SST-2 task (the overall trend is the same for the other layers and tasks). As shown in the figure, quantization significantly increases the cover length ratio and the ranking loss, indicating that the relative ratings of the attention are seriously distorted. The attention-score loss helps reduce such distortion, yet spikes still exist. In contrast, attention-map loss successfully suppresses the spikes, maintaining the relative importance of the attention map. More details about cover length ratio and ranking loss are described in Appendix. A.1.

To further understand the impact of the objectives of KD on the QAT accuracy, we conducted the temperature sweep of KL-Div. Since the gradients of KL-Div loss can be simplified into the gradients of MSE loss when the temperature is sufficiently large (Kim et al., 2021b), we can manage the behavior of KL-Div loss through sweeping the temperature value (τ), where $\tau=1$ and $\tau = \inf$ resemble the attention-map and attention-score losses, respectively. Table 1 shows the QAT accuracy of BERT-Base on CoLA and STS-B with varying τ.

As shown in the table, the accuracy of the quantized model increases as the loss term becomes similar to the attention-map loss. Such performance improvement supports our understanding that 1) label matching is crucial for compensating QAT on SA-GEN, and 2) the attention-map loss is more effective for label matching.

| Task | KL Divergence Temperature Hyper-Parameter (τ) | | | | | |
	KL Div	$\tau=1$	$\tau=5$	$\tau=10$	$\tau=20$	MSE
CoLA	**50.76**	50.76	49.11	47.51	47.19	47.51
STS-B	**87.78**	87.69	87.20	87.19	87.29	87.55

Table 1: Attention based KD-QAT with KL Div Temperature Hyper Parameter sweeping on CoLA and STS-B tasks with BERT-Base (Each experiment is repeated 5 times.)

4 KD for QAT on Large Transformers

We extend the investigation of KD techniques to QAT on large transformer models. In this section, we first reveal the limitation of the attention-map loss due to the task-dependent characteristics. Then we propose a new KD loss, the attention-output loss, to address this challenge. Lastly, we propose a combination of the two losses to handle task-dependent characteristics.

4.1 Task-Dependent Characteristics

Although the same pre-trained models are employed for the downstream fine-tuning, the characteristics of attention vary depending on the tasks (Kovaleva et al., 2019). Motivated by the discussions of (Bondarenko et al., 2021) that outliers in activations of residual connections (i.e., SA-PROP) arrange specific attention patterns, we examine these outliers via min-max curves at attention output to understand task-dependent characteristics for quantization.

Fig. 5 plots the dynamic range of the attention output (Y_l) across the input tokens. There are two representative cases: Case-1) the task with distinct attention values (especially for special tokens) such as RTE, and Case-2) the task with homogeneous attention values such as SST-2. Each case's overall attention characteristics are intensified as the model size increases. For example, distinct features of RTE's attention become more drastic on BERT-Large.

Since the quantization clamps and coarsely represents the values, it is challenging to maintain the distinct attention for the tasks in Case-1. As dis-

cussed in Sec. 3.2, in the case of BERT-Base, the attention-map loss was capable of recovering the disparity in the attention (Fig. 5a-Top). However, as shown in Fig. 5a-Below, the attention-map loss failed to adjust the attention in the case of BERT-Large.

We conjecture that the attention-map loss fails due to the increased number of layers of BERT-Large. We adopt the analysis framework of (Kobayashi et al., 2020) to separately analyze the layer-wise behavior of SA-GEN and SA-PROP. Fig. 6 plots the average distance of SA-GEN and SA-PROP of a special token *[SEP]* from the teacher for RTE (Case-1) and SST-2 (Case-2) with BERT-Large. Note that the attention-map loss suppresses the distance in SA-GEN. This suppression of SA-GEN deteriorates the attention output (c.f., the attention-map loss is effective for SST-2). With many layers, quantization along SA-PROP fails KD with attention-map loss to recover distinctive attention. Therefore, we need a new KD loss to handle disruption from SA-PROP.

4.2 Attention Output Loss

Observations from Fig. 6 imply that SA-PROP becomes a disruption source for QAT of BERT-Large on Case-1 tasks. One way to suppress the quantization error along SA-PROP is to apply KD directly to SA-PROP. Therefore, we devise a new KD loss, the attention-output loss as follows:

$$\mathcal{L}_{output} = \sum_{l=0}^{L-1} \text{MSE}(\mathbf{Y}_l^T, \mathbf{Y}_l^S). \quad (7)$$

The benefits of the attention-output loss are apparent. As shown in Fig. 5a, the attention output with the attention-output loss follows the distinctive attention of the full-precision teacher. We can understand the mechanism of the attention-output loss via Fig. 6; the attention-output loss allows modification of SA-GEN to adjust the attention map so that the resulting attention output matches better with the teacher. Note that the change in SA-GEN occurs at the upper layers of the Transformer models; thus, the attention-output loss is more beneficial for large Transformer models.

To further understand the task-dependent characteristics, we empirically observe the attention-output loss's impact on the attention map's self-attention probability. To quantify the modification in the attention map, we introduce the ranking ratio, defined as a ranking of the attention probability of
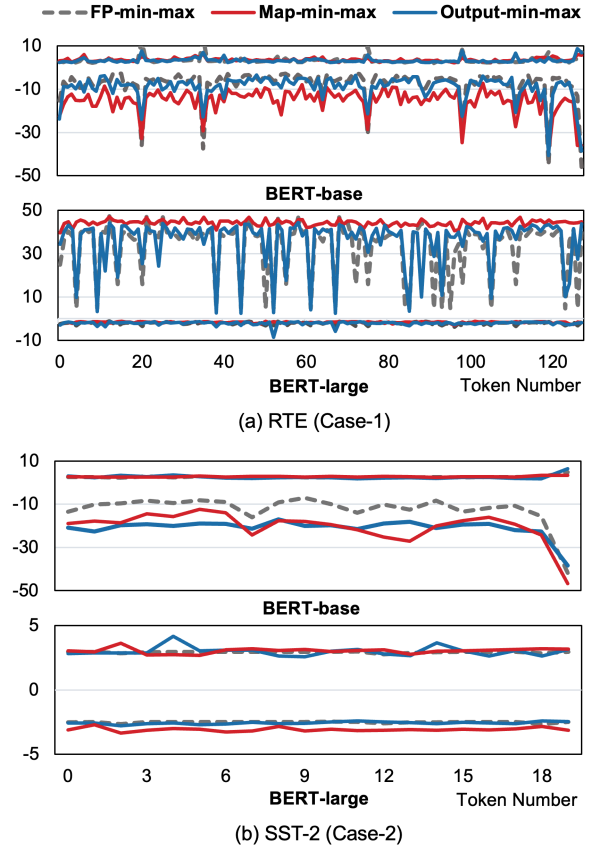

Figure 5: Comparison of per-token dynamic ranges of attention output (Y_l) between BERT-Base (top) and BERT-Large (bottom) for RTE and SST-2. Each pair of curves delineates min-max values at the token's attention output. FP-min-max and Map/Output-min-max correspond to the min-max curves of the teacher model and the student with the attention map/output loss, respectively.

an individual token normalized by the sequence length. Fig. 7 tracks the ranking ratio of the selected tokens of full-precision teacher and quantized student models per each head. In the case of RTE (i.e., Case-1 task), QAT with the attention-output loss exhibits rapid changes in ranking in a specific direction toward reduced attention-output loss. In the case of SST-2, however, the situation is very different; rankings of the selected tokens change significantly regardless of KD for QAT. Thus, KD on the attention output cannot drive the rankings in any meaningful direction. These observations confirm the importance of considering task-dependent characteristics for successful KD for QAT.

4.3 Unified Attention-Map and Output Loss

Considering task-dependent attention characteristics of BERT-Large, we further explore the potential of unifying the attention-map and output
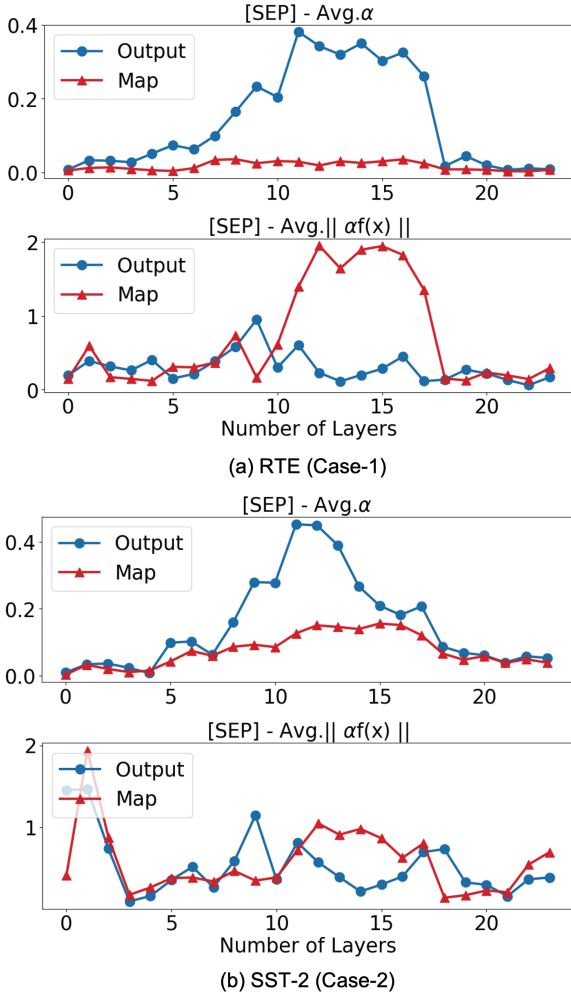
(a) RTE (Case-1)

(b) SST-2 (Case-2)

Figure 6: Average distance of SA-GEN(self-attention probability - α) and SA-PROP(self-attention propagation - $f(x)$) from the teacher model in two tasks (RTE, SST-2) with BERT-Large.

(a) RTE (Case-1)

(b) SST-2 (Case-2)

Figure 7: Ranking ratio per head (Last three layer's 48 heads) on the SST-2 and RTE task with BERT-Large. Left: Teacher Model, Right: Student Model

5 Experiments

5.1 Experimental Setup

We employ three Transformer encoder models (BERT-Base, BERT-Large, ULM-Encoder-Large) to evaluate the proposed KD methods. BERT (Devlin et al., 2019) consists of Transformer encoder layer, finetuned for GLUE downstream tasks (Devlin et al., 2019). ULM-Encoder-Large (Seo et al., 2022) is a Korean language model based on T5 (Raffel et al., 2019), finetuned for KLUE downstream tasks (Park et al., 2021a).

The configuration of each model is as follows:

1. **BERT-Base.** It is a 12-layer Transformer encoder with a hidden dimension of 768 using 12 attention heads and contains about 110M parameters.

2. **BERT-Large.** It is composed of 24 Transformer encoder layers, and uses a hidden dimension of 1024 with 16 attention heads. This model contains about 340M parameters.

3. **ULM-Encoder-Large.** It also has the same configuration as BERT-large except for feed-forward dimension, which is 2816 for ULM-Encoder-Large while BERT-Large has 4096. It contains about 280M parameters.

We initiate QAT from the task-specific fine-tuned models. Our experiments were performed on A6000 GPUs. Our implementation is based on the TernaryBERT PyTorch codebase.[1] All embed-

losses for QAT. Note that the preference between the attention-map and output losses varies according to the model size (e.g., BERT-Base vs. Large) and tasks (Case-1 vs. Case-2). As for exploration, we formulate a unified attention-map and output loss with γ as a mixing parameter as follows:

$$\mathcal{L}_{SM_1} = \mathcal{L}_{map} + \gamma \mathcal{L}_{output},$$
$$\mathcal{L}_{SM_2} = \mathcal{L}_{output} + \gamma \mathcal{L}_{map}, \qquad (8)$$
$$\text{where } \gamma \in \{0.1, 0.2, 0.3, \dots, 0.9\}.$$

As will be discussed in Sec.5.2, the unified loss can boost the accuracy of the best performing KD loss (either the attention-map or output loss). As applying this unified loss in KD-QAT, we identified that every tasks has its own score favorable mixing parameters which shows task-dependent characteristics. Detailed mixing parameter information for each task is in Appendix. A.3.
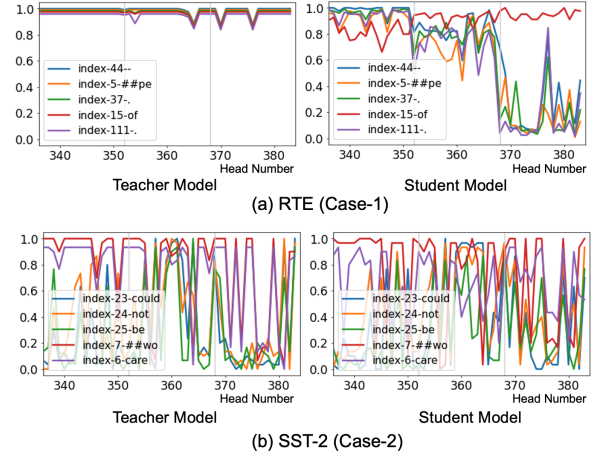
---

[1] https://github.com/huawei-noah/Pretrained-Language-Model/tree/master/TernaryBERT

GLUE Task (Dataset)	RTE† (2.5k)	CoLA† (8.5k)	STS-B† (5.7k)	SST-2* (67k)	QNLI* (108k)	MNLI* (393k)	QQP* (364k)	MRPC (3.5k)	AVG
Full-Prec	73.28	58.04	89.24	92.09	91.32	84.37	89.30	87.77	83.39
Baseline	68.53 $_{\pm1.69}$	49.61 $_{\pm0.79}$	87.55 $_{\pm0.14}$	92.01 $_{\pm0.29}$	90.65 $_{\pm0.05}$	84.21 $_{\pm0.10}$	89.06 $_{\pm0.40}$	**88.58** $_{\pm0.40}$	81.28
Map	70.39 $_{\pm0.78}$	50.40 $_{\pm1.03}$	**87.78** $_{\pm0.15}$	92.13 $_{\pm0.22}$	**90.98** $_{\pm0.17}$	84.31 $_{\pm0.10}$	89.22 $_{\pm0.40}$	88.07 $_{\pm0.40}$	81.66
Output	70.65 $_{\pm1.27}$	49.05 $_{\pm0.50}$	87.77 $_{\pm0.14}$	92.13 $_{\pm0.22}$	90.58 $_{\pm0.07}$	84.24 $_{\pm0.01}$	89.17 $_{\pm0.20}$	87.01 $_{\pm0.43}$	81.33
Map+Output	**71.68** $_{\pm1.19}$	**50.50** $_{\pm0.45}$	87.73 $_{\pm0.16}$	**92.39** $_{\pm0.18}$	90.91 $_{\pm0.14}$	**84.33** $_{\pm0.06}$	**89.28** $_{\pm0.10}$	88.18 $_{\pm0.53}$	**81.87**

Table 2: BERT-Base: Performance of KD-QAT Results on GLUE benchmark (8-bit activation and ternary weight quantization, the compression rate of quantized BERT-Base is 14.9x). Small dataset (under 10k) tasks are repeated 5 times; the others are repeated 3 times. $\dagger$ and $\star$ indicate Case-1 and Case-2 GLUE tasks respectively.

GLUE Task (Dataset)	RTE† (2.5k)	CoLA† (8.5k)	STS-B† (5.7k)	SST-2* (67k)	QNLI* (108k)	MNLI* (393k)	QQP* (364k)	MRPC (3.5k)	AVG
Full-Prec	70.39	60.31	89.83	92.32	92.29	86.49	89.55	88.43	83.70
Baseline	65.02 $_{\pm1.40}$	52.87 $_{\pm0.99}$	88.75 $_{\pm0.09}$	91.82 $_{\pm0.22}$	91.87 $_{\pm0.15}$	85.70 $_{\pm0.17}$	89.29 $_{\pm0.07}$	**89.26** $_{\pm0.54}$	81.84
Map	66.42 $_{\pm0.75}$	53.16 $_{\pm0.53}$	88.65 $_{\pm0.11}$	92.20 $_{\pm0.30}$	91.93 $_{\pm0.13}$	86.10 $_{\pm0.13}$	**89.53** $_{\pm0.07}$	88.67 $_{\pm0.37}$	81.28
Output	**69.50** $_{\pm1.20}$	**54.71** $_{\pm0.71}$	**89.10** $_{\pm0.08}$	92.13 $_{\pm0.26}$	91.92 $_{\pm0.13}$	86.22 $_{\pm0.05}$	89.44 $_{\pm0.09}$	88.75 $_{\pm0.71}$	**82.72**
Map+Output	68.83 $_{\pm1.45}$	54.69 $_{\pm1.08}$	88.85 $_{\pm0.15}$	**92.30** $_{\pm0.11}$	**92.16** $_{\pm0.15}$	**86.36** $_{\pm0.06}$	89.48 $_{\pm0.06}$	88.64 $_{\pm0.79}$	82.66

Table 3: BERT-Large: Performance of KD-QAT Results on GLUE benchmark (8-bit activation and ternary weight quantization, the compression rate of quantized BERT-Large is 15.4x). Small dataset (under 10k) tasks are repeated 5 times; the others are repeated 3 times. $\dagger$ and $\star$ indicate Case-1 and Case-2 GLUE tasks respectively.

ding and weight parameters are ternarized and the activations are quantized to 8-bit for QAT. we use layer-wise ternarization for weights in Transformer layers while row-wise for the word embedding, same as TernaryBERT (Zhang et al., 2020). Also, all the experiments are repeated 5 times, unless stated otherwise.

For performance comparison, we consider the following KD options:

- **Baseline.** The standard TernaryBERT with the attention-score and Transformer output loss along with KD on soft labels.

- **Map.** Use the attention-map loss instead of the attention-score loss of TernaryBERT.

- **Output.** Use the attention-output loss instead of the attention-score loss of TernaryBERT.

- **Map+Output.** Use the unified attention-map and output loss instead of the attention-score loss of TernaryBERT.

5.2 Experiments on BERT-Base and Large

Tables 2 and 3 show the result on the development set across the GLUE benchmark. Notable observations are summarized as follows:

- The GLUE tasks can be categorized into two cases. Case-1($\dagger$): RTE, CoLA, STS-B. Case-2($\star$): SST-2, QNLI, MNLI, QQP.

- In the case of BERT-Base, attention-map loss benefits all the tasks in Case-1 and Case-2, whereas attention-output loss is ineffective.

- In the case of BERT-Large, the attention-map loss is marginally helpful for Case-1 and Case-2, while the attention-output loss significantly boosts the accuracy of Case-1 tasks.

- Overall, the unified loss facilitates QAT accuracy, except for BERT-Large on Case-1 tasks (in which the attention-output loss works the best).

- MRPC is a corner case; the QAT accuracy often outperforms the Full-Precision accuracy, implying that quantization noise regularizes the model favorably for this task.

5.3 Experiments on ULM-Encoder-Large

Table 4 summarizes the average results three times of evaluating our KD methods for QAT on ULM-Encoder-Large. Overall, ULM-Encoder-Large is quite robust to quantization, but our KD methods surpass the baseline (= TernaryBERT). More specifically, the attention-map loss is more effective on

Task (Dataset)	KLUE-TC (45k)	KLUE-STS (11k)	NSMC (150k)	AVG
Full-Prec	85.76	92.11	91.87	89.91
Baseline	85.56 ±0.08	91.04 ±0.10	91.13 ±0.04	89.24
Map	85.41 ±0.10	**91.44** ±0.23	91.24 ±0.10	89.36
Output	**85.63** ±0.23	91.03 ±0.11	91.39 ±0.15	89.35
Map + Output	85.57 ±0.21	91.11 ±0.14	**91.65** ±0.12	**89.44**

Table 4: ULM-Large: Performance of KD-QAT Results on KLUE and NSMC dev dataset

GLUE Task	RTE[†]	CoLA[†]	STS-B[†]	SST-2[⋆]	QNLI[⋆]
Full-Prec	70.39	60.31	89.83	92.32	92.29
Map	66.42	53.16	88.65	**92.20**	**91.93**
MHA loss	66.78	54.01	88.69	92.08	91.84
MHA loss + Residual	**69.50**	**54.71**	**89.10**	92.13	91.92

Table 5: Ablation study on the attention-output loss decomposing its sources into $MHA(\mathbf{X}_l)$ and residual path $(\mathbf{X}_l)$ with BERT-Large on the GLUE tasks. † and ⋆ indicate Case-1 and Case-2 GLUE tasks respectively.

the KLUE-STS task, while the output loss outperforms the map loss in KLUE-TC and NSMC, as shown in the table. Furthermore, the unified loss achieves the best accuracy on NSMC. Therefore, the proposed KD losses could improve the accuracy of the baseline QAT method.

5.4 Ablation Study

In Sec. 4.2, we proposed attention-output loss to suppress the quantization error along the SA-PROP. As shown in Fig. 1 and defined in Eq. 4, attention-output loss integrates two sources of SA-PROP: $MHA(\mathbf{X}_l)$ and residual connection $(\mathbf{X}_l)$. We investigate attention-output loss's effectiveness by employing one of its two parts solely as a loss function objective: $MHA(\mathbf{X}_l)$ (We call this loss function MHA loss). Specifically, MHA loss uses $MHA(\mathbf{X}_l)$ as a objective of loss function instead of $\mathbf{Y}_l$ in Eq. 7.

Table. 5 shows that the MHA loss method improves performance marginally in Case-1 tasks. When the residual connection is added to the MHA loss objective (MHA loss + Residual in Table. 5), which is equivalent to attention-output loss, the performance of all tasks increases. (especially in Case-1 GLUE tasks). These observations indicate that incorporating residual connection as an objective of attention-output loss is significant in recovering disruption of SA-PROP under the quantization.

6 Conclusion

In this work, we investigate the mechanism of Knowledge distillation (KD) for QAT of large

Transformers. We propose two KD methods, attention-map, and attention-output losses, to improve the recovery of the self-attention information. The experimental results on various Transformer encoder models demonstrate that the proposed KD methods and their combination achieve state-of-the-art accuracy for QAT with sub-2-bit weight quantization. Our code is available at https://github.com/MarsJacobs/kd-qat-large-enc.

7 Limitation

This work investigates how KD works for QAT on Transformer Encoders. Although the analysis techniques employed in this work reveal many exciting insights, a more theoretical analysis of the impact of quantization under KD would be highly appreciated. Also, we explore the potential of unifying the two proposed KD techniques; incorporating automatic balancing of the two (or more) KD losses would be an interesting future research direction.

Acknowledgement

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.

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*.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2021. Understanding and overcoming the challenges of efficient transformer quantization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7947–7969, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

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

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.

Jing Jin, Cai Liang, Tiancheng Wu, Liqin Zou, and Zhiliang Gan. 2021. Kdlsq-bert: A quantized bert combining knowledge distillation with learned step size quantization. *arXiv preprint arXiv:2101.05938*.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2021a. I-bert: Integer-only bert quantization. *International Conference on Machine Learning (Accepted)*.

Taehyeon Kim, Jaehoon Oh, Nakyil Kim, Sangwook Cho, and Se-Young Yun. 2021b. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. *CoRR*, abs/2105.08919.

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.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. 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 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Zheng Li, Zijian Wang, Ming Tan, Ramesh Nallapati, Parminder Bhatia, Andrew Arnold, Bing Xiang, and Dan Roth. 2022. Dq-bart: Efficient sequence-to-sequence model via joint distillation and quantization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 203–211.

Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. 2021. Fq-vit: Fully quantized vision transformer without retraining. *arXiv preprint arXiv:2111.13824*.

Yuanxin Liu, Fandong Meng, Zheng Lin, Weiping Wang, and Jie Zhou. 2021a. Marginal utility diminishes: Exploring the minimum knowledge for BERT knowledge distillation. *CoRR*, abs/2106.05691.

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*.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Namuk Park and Songkuk Kim. 2022. How do vision transformers work? In *International Conference on Learning Representations*.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Ji Yoon Han, Jangwon Park, Chisung Song, Junseong Kim, Youngsook Song, Taehwan Oh, et al. 2021a. Klue: Korean language understanding evaluation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won-Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Tae Hwan Oh, Joohong Lee, Juhyun Oh, Sungwon Lyu, Younghoon Jeong, Inkwon Lee, Sangwoo Seo, Dongjun Lee, Hyunwoo Kim, Myeonghwa Lee, Seongbo Jang, Seungwon Do, Sunkyoung Kim, Kyungtae Lim, Jongwon Lee, Kyumin Park, Jamin Shin, Seonghyun Kim, Eunjeong Lucy Park, Alice Oh, Jung-Woo Ha, and Kyunghyun Cho. 2021b. KLUE: korean language understanding evaluation. *CoRR*, abs/2105.09680.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683.

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*.

Youngkyung Seo, Jehoon Lee, TaeHyeong Kim, Junseok Koh, Hwijung Ryu, and Du-Seong Chang. 2022. External Knowledge-based Language Model Enhancement. GTC.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *CoRR*, abs/1804.04235.

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.

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, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. 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.

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.

Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020. 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.

A   Appendix

A.1   Cover Length Ratio and Ranking Loss

As shown in Sec. 3.2, cover length ratio and ranking loss are metrics indicating how much the relative importance within the attention map has deviated under the quantization. To obtain the cover length ratio, first, sort the student and teacher's attention map in probability order. Then we can get the teacher map's Top-K tokens that receive the most attention in the teacher attention map. After that, we find out how many tokens we need to look at to get all the teacher map's Top-K tokens from the sorted student map. The number of tokens that each token has to look at for covering the teacher map's Top-K tokens is called cover length, and we normalize cover length by sequence length. We call this metric cover length ratio.

(Liu et al., 2021b) introduced pairwise ranking loss to keep the relative order of attention values. Pairwise ranking loss indicates how different the order of attention importance is between two attention maps. With any of the two tokens, if the order of the two tokens in the student map is different from that in the teacher map, we add the difference between the values of the two tokens to the loss.

$$
\begin{aligned}
\mathcal{L}_{ranking}^{h} \\
= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \Phi((AM_i^S - AM_j^S) \\
\cdot sign(AM_i^T - AM_j^T)), \quad (9) \\
\mathcal{L}_{ranking} = \sum_{h=1}^{N_H} \mathcal{L}_{ranking}^{h},
\end{aligned}
$$

where h is an index of attention head.

A.2   Visualization of Attention Map

We compare the self-attention map of fine-tuned full-precision BERT-base model and the quantized model on the RTE task. Fig. 8 shows self-attention maps from the 3rd Transformer layer (8th head) of BERT-Base. Note that applying quantization distorts the self-attention map of the teacher severely. As shown in the figure, attention-map loss successfully recovers the self-attention map of the teacher, whereas TernaryBERT fails to capture some distinctive features.
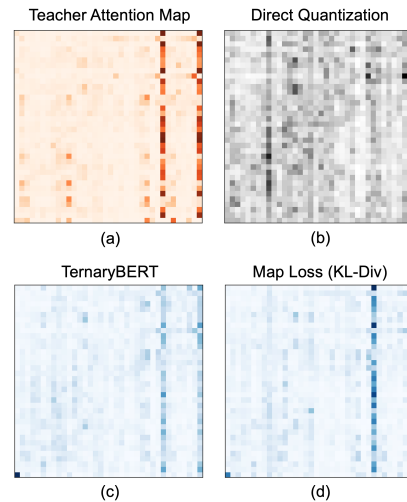

Figure 8: Visualization of self-attention map with BERT-Base over RTE Task (a) Teacher self-attention map (b) After quantization without QAT (c) After TernayBERT KD-QAT (d) Attention-map loss

A.3   Unified Attention-Map and Output Loss Exploration

As mentioned in Sec. 4.3, we explore the effectiveness of unifying the attention map and output losses for QAT. We conducted the experiments by fixing one of two losses, attention-map, and output loss, and changing the mixing parameter γ for another loss according to Eq. 8. Table 7 and 8 show the best results changing mixing parameter γ in BERT-Base and BERT-Large. Overall, maintaining attention-map loss (case SM_1) perform better than opposite cases (case SM_2). To show the degree of effect according to mixing parameter γ, we summarize the γ showing the best result for each case in Table 6. The table shows that every task has its own score favorable mixing parameter γ. These trends can be connected to the task-dependent attention characteristics in Sec 4.1.

| GLUE | BERT-Base | | BERT-Large | |
Task	SM_1	SM_2	SM_1	SM_2
RTE	0.6	0.4	0.1	0.9
CoLA	0.4	0.2	0.1	0.5
STS-B	0.7	0.2	0.3	0.9
MRPC	0.1	0.1	0.1	0.9
SST-2	0.2	0.7	0.3	0.1
QNLI	0.8	0.8	0.8	0.6
MNLI	0.5	0.1	0.8	0.7
QQP	0.1	0.1	0.5	0.6

Table 6: Results of Map+Output mixing parameter γ exploration

GLUE Task	RTE	CoLA	STS-B	MRPC	SST-2	QNLI	MNLI	QQP	AVG
Full-Prec	73.28	58.04	89.24	87.77	92.09	91.32	84.37	89.30	83.39
SM_1	**71.68** ±1.19	**50.50** ±0.45	**87.73** ±0.16	88.18 ±0.53	**92.39** ±0.18	90.90 ±0.14	**84.33** ±0.06	**89.28** ±0.10	81.87
SM_2	71.48 ±0.96	50.10 ±1.02	87.71 ±0.09	**88.22** ±0.39	92.32 ±0.11	**90.91** ±0.07	84.24 ±0.04	89.22 ±0.08	81.83

Table 7: BERT-Base Map+Output performance results on GLUE benchmark. Small dataset (under 10k) tasks are repeated 5 times; the others are repeated 3 times.

GLUE Task	RTE	CoLA	STS-B	MRPC	SST-2	QNLI	MNLI	QQP	AVG
Full-Prec	70.39	60.31	89.83	88.43	92.32	92.29	86.49	89.55	83.70
SM_1	**68.83** ±1.45	**54.69** ±1.08	88.85 ±0.15	**88.64** ±0.79	**92.31** ±0.11	92.16 ±0.15	86.32 ±0.03	**89.48** ±0.05	82.64
SM_2	68.23 ±1.06	54.44 ±0.46	**88.93** ±0.08	88.24 ±0.26	92.20 ±0.30	92.16 ±0.07	**86.36** ±0.06	89.45 ±0.03	82.46

Table 8: BERT-Large Map+Output performance results on GLUE benchmark. Small dataset (under 10k) tasks are repeated 5 times; the others are repeated 3 times.

A.4 Experimental Setup

Datasets

We evaluate our method on all datasets of the GLUE benchmark (Wang et al., 2018) for BERT, and two datasets of the KLUE (Park et al., 2021b), which is one of the datasets to evaluate the natural language understanding capability of Korean language models, and sentiment analysis NSMC[2] datasets. Details are as follows.

1. **GLUE.** The General Language Understanding Evaluation is a collection of resources for training, evaluating, and analyzing natural language understanding systems.

2. **KLUE-TC.** The KLUE Topic Classification is a single sentence classification task, and it classifies which topic the input sentence belongs to among the 7 representative topics. We averaged the accuracy and F1 score as the metric.

3. **KLUE-STS.** The KLUE Semantic Textual Similarity is to measure the degree of semantic similarity between two Korean sentences. We averaged the Pearson Correlation Coefficient (PCC) and Spearman Correlation Coefficient (SCC) to measure the performance.

4. **NSMC.** The NAVER Sentiment Movie Corpus has a collection of movie reviews scraped from NAVER Movies[3], including an annotation of whether the evaluation of the movie is positive or negative.

Training Settings

For evaluating our methods, we use batch size 16 for CoLA and 32 for other GLUE 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 3 epochs. The dropout probability was always kept at 0.1. For an optimizer, we use BertAdam[4], which is a variant of Adam. For ULM-Large, we train for 10 epochs using AdamW (Loshchilov and Hutter, 2017) for KLUE-TC and Adafactor (Shazeer and Stern, 2018) for KLUE-STS and NSMC. We empirically find the best hyperparameter setting for each task in the following choices:

- **Batch size:** 16, 32, 64

- **Learning rate:** 1e-5, 2e-5, 5e-5

---

[2]https://github.com/e9t/nsmc
[3]https://movie.naver.com/movie/point/af/list.naver

[4]https://github.com/huggingface/transformers/blob/v0.6.2/pytorch_pretrained_bert/optimization.py