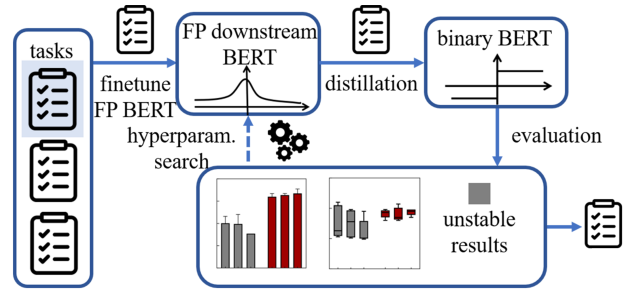
BiPFT: Binary Pre-trained Foundation Transformer with Low-Rank Estimation of Binarization Residual Polynomials

Xingrun Xing[1,2,3], Li Du[3], Xinyuan Wang[4], Xianlin Zeng[4], Yequan Wang[3], Zheng Zhang[3*], Jiajun Zhang[1*]

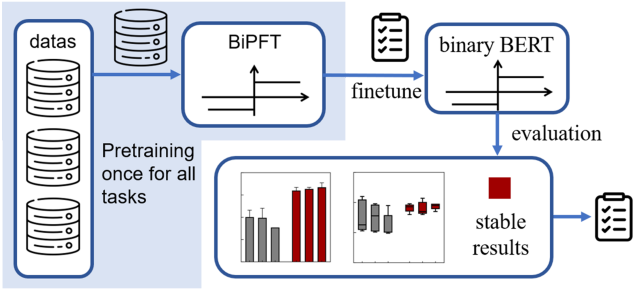
[1]Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]Beijing Academy of Artificial Intelligence
[4]Beihang University
xingxingrun2023@ia.ac.cn, duli@baai.ac.cn, buaa42wxy@gmail.com, zengxianlin@buaa.edu.cn, tshwangyequan@gmail.com, zhangz.goal@gmail.com, jjzhang@nlpr.ia.ac.cn

Abstract

Pretrained foundation models offer substantial benefits for a wide range of downstream tasks, which can be one of the most potential techniques to access artificial general intelligence. However, scaling up foundation transformers for maximal task-agnostic knowledge has brought about computational challenges, especially on resource-limited devices such as mobiles. This work proposes the first Binary Pretrained Foundation Transformer (BiPFT) for natural language understanding (NLU) tasks, which remarkably saves 56× operations and 28× memory. In contrast to previous task-specific binary transformers, BiPFT exhibits a substantial enhancement in the learning capabilities of binary neural networks (BNNs), promoting BNNs into the era of pre-training. Benefiting from extensive pretraining data, we further propose a data-driven binarization method. Specifically, we first analyze the binarization error in self-attention operations and derive the polynomials of binarization error. To simulate full-precision self-attention, we define binarization error as binarization residual polynomials, and then introduce low-rank estimators to model these polynomials. Extensive experiments validate the effectiveness of BiPFTs, surpassing task-specific baseline by 15.4% average performance on the GLUE benchmark. BiPFT also demonstrates improved robustness to hyperparameter changes, improved optimization efficiency, and reduced reliance on downstream distillation, which consequently generalize on various NLU tasks and simplify the downstream pipeline of BNNs. Our code and pretrained models are publicly available at https://github.com/Xingrun-Xing/BiPFT.



(a) Previous task-specific pipeline to train a binary transformer.



(b) Our pretrain-finetuning pipeline with the binary foundation model.

Figure 1: Comparison of training pipelines for binary transformers. FP indicates full-precision. For downstream tasks, finetuning BiPFT replaces previous task-specific pipelines.

Introduction

In recent years, pre-trained foundation models (PFM) (OpenAI 2023) have demonstrated impressive emergent intelligence phenomena in various fields such as natural language processing (Touvron et al. 2023) and computer vision (Kirillov et al. 2023; Wang et al. 2023). As the model size and pre-training data increase, task-agnostic knowledge from pretraining effectively generalizes to downstream tasks with small datasets or open scenes. In natural language understanding (NLU) tasks, BERT (Devlin et al. 2018; He, Gao, and Chen 2023), which uses the transformer encoder architecture and the bi-directional masked prediction training, is widely applied. However, the self-attention (Vaswani et al. 2017a; Carlini et al. 2023) and MLP layers in BERT involve substantial floating-point operations and memory consumption. How to get a compact pretrained foundation model in computationally limited settings, such as inference at mobile devices, has become a problem of significant value.

This work aims to propose the first 1-bit pretrained foundation model for NLU tasks with a BERT like architecture.

*Corresponding Author

Recently, compression methods for BERT include model pruning (Gordon, Duh, and Andrews 2020; Zhao and Wressnegger 2023), distillation (Sun et al. 2020; Ding et al. 2023), and quantization (Kim et al. 2021; Castano et al. 2023). Model quantization achieves a high degree of model compression without changing the model architecture or the number of parameters. Notably, 1-bit model quantization is an extreme case of low-bit quantization. Unlike other low-bit models, binary neural networks (BNNs) (Courbariaux et al. 2016; Xu et al. 2023) directly utilize the underlying XNOR and popcount operations instead of numerical ones, thereby achieving *super-linear benefits of bit-width*. Compared to full-precision (FP) model inference, binary models save up to 64$\times$ operations, 32$\times$ memory, and between 100 to 1000$\times$ energy consumption (Courbariaux et al. 2016), which are necessary for modern large pretrained foundation models.

Current binary transformers perform binarization on specific tasks. Due to their extremely low bit-width, these binary transformers face significant optimization challenges. To address this issue, prior binary BERTs rely on optimization techniques such as distillation from the full-precision (FP) teacher and hyperparameter tuning. As illustrated in Fig. 1 (a), the typical pipeline is complex: it begins by training a FP teacher for the given task, followed by initializing and distilling the binary BERT using this FP teacher. Because of unstable optimization, a hyperparameter search is usually necessary. We want to ask the question *whether a 1-bit BERT, with initialization and distillation from the downstream FP model, is able to achieve similar performance even without pretraining?* We build a strong binary transformer baseline and conduct extensive experiments in different training settings (including the distillation, learning rate, batch size, etc.) and want to find the keypoints to influence performance. These non-trivial experiments indicate the weakness of task-specific binary transformers:

Unstability to hyperparameters. Our experiments show that task-specific binary BERTs have a large performance variance to different batch sizes and learning rates. The performance heavily relies on hyperparameters tuning and often requires a small batch size and long training time.

Weakness of learning capabilities. Existing task-specific binary BERTs (Qin et al. 2022; Liu et al. 2022) also heavily rely on the distillation from FP teacher. When we replace the distillation loss with direct training loss, the average performance drops by 13.9% on the GLUE benchmark.

This phenomenon suggests the necessity of directly training 1-bit foundational models, rather than initializing a binary model with its 32-bit task-specific counterpart.

We propose the first Binary Pretrained Foundation Transformer, termed BiPFT, promoting BNNs into the era of pretraining. We start with building a general baseline architecture for binary transformers. Based on this architecture, we then pretrain a binary foundation model named BiPFT-A and evaluate the impact of pretraining for BNNs. During pretraining, we followed the standard masked language model (MLM) and next sentence prediction (NSP) tasks used in FP BERTs. In addition, a task-agnostic distillation is also attached to speed up pretraining. In contrast to task-specific distillation, task-agnostic distillation doesn't complicate the downstream pipeline. After pretraining, the learning capabilities of binary transformers are improved significantly, which enables binary transformers directly finetuned without distillation and hyperparameter tuning. As shown in Fig. 1 (b), given a new task, binary pretrained foundation transformers only require straightforward finetuning, eliminating the complexity of previous downstream pipelines. Experimental results show that under fair comparison, BiPFT-A improves 13.9% average performance on the GLUE benchmark compared with the baseline model without binary pretraining. Even when compared to the baseline that employs additional hyperparameter tuning and distillation, BiPFT-A still surpasses it by 1.1% with simple finetuning.

With the pretraining phase, we rethink how to effectively binarize self-attention. Previous works mainly focus on empirically designing more accurate binary operations. For example, BiBERT (Qin et al. 2022) designs a Bi-Attention operation to simulate FP self-attention; BiT (Liu et al. 2022) designs the $\{0, 1\}$ binarization level and elastic binarization functions to better simulate FP activations. In contrast to performing binarization in downstream tasks with very limited data previously, binary pretrained foundation models perform binarization in the pretraining phase, making it possible to use data-driven and data-hungry binarization methods. Specifically, we analyze the binarization error in self-attention operations and derive the polynomials of binarization error. To simulate full-precision self-attention, we indicate binarization error as binarization residual polynomials and then introduce low-rank estimators to model binarization residual polynomials. Low-rank estimators are fully trained in pretraining, while estimators generalize to data-limited tasks effectively in downstream. We add the aforementioned residual polynomial estimators to BiPFT-A and name the new model as BiPFT-B. Experimental results indicate that BiPFT-B enhances performance on GLUE by an additional 1.6% compared to BiPFT-A.

The contributions of this paper are as follows:

- We propose the first binary pretrained foundation model and successfully train BNNs throughout the pretraining and finetuning phases.
- We propose a data-driven binarization method for self-attention by estimating binarization residual polynomials, further improving binary foundation models.
- We release binary foundation transformers for NLU tasks. Finetuning on this foundation model for downstream tasks significantly simplifies the training process of BNNs, yielding more robust and accurate results.

Related Work

Most studies of binary neural networks (He et al. 2023; Kunes et al. 2023; Xing et al. 2022a) focus on convolutional neural networks in the computer vision field. BNNs are first proposed by directly binarizing both activations and weights to the bit-width of 1 and estimating gradient using straight-through estimators (STE) (Courbariaux et al. 2016). However, vanilla BNNs encounter performance drop in large-scale datasets. Many works improve BNN performance from different perspectives, including model architecture, binary

parameter optimization and binarization strategy (Martinez et al. 2020). For example, BiRealNet (Liu et al. 2018) and CP-NAS (Li'an Zhuo et al. 2020) revise more efficient binary network architectures. XNOR-Net and Siman (Lin et al. 2022) focus on optimizing binarization error. ReActNet (Liu et al. 2020) revise binarization and activation functions to improve model capacity. More recently, BCDNet (Xing et al. 2022b) introduce MLP (Chen et al. 2023) architecture to BNNs and achieve high performance.

In the natural language processing field, BinaryBERT (Bai et al. 2021), BiBERT (Qin et al. 2022) and BiT (Liu et al. 2022) binarize full-precision BERT model in specific tasks. TBT (Liu et al. 2023) and DQ-BART (Li et al. 2022) distill binary and low-bit generation models in specific tasks. However, previous binary transformers heavily rely on task-specific distillation. There is no foundation model directly pretrained with binary parameters and activations.

Compared with post-training quantization (PTQ), BNNs adopt quantization-aware training (QAT). Although some PTQ methods are well-known for language models such as OPTQ (Frantar et al. 2022) and SmoothQuant (Xiao et al. 2023), they cannot achieve 1-bit width.

Methodology

Build Binary Baseline Architecture

We define a baseline model as the benchmark of binary transformers and introduce pretraining of the baseline model in the next section. Existing task-specific binary transformers often use different binarization, training and evaluation methods, making it challenging to compare the general performance. To build a general baseline, we follow the binarization design of BiTs as much as possible, while replacing their specific training and evaluation settings with common ones. The differences between our baseline and BiTs are shown in the Appendix A of our extended version (Xing et al. 2023) in detail. We briefly introduce the basic binary operations in the baseline as follows:

Binary linear. Binary linear layers compose the most basic operations in a binary transformer, which indicates binarizing both weights and activations to the bit-width of 1. In forward propagation, FP weights $\mathbf{W}$ and activations $\mathbf{A}$ are initially binarized to $\mathbf{W_B}$ and $\mathbf{A_B}$ using the binarization function $\mathbf{Q_B}$. Consequently, linear layers can be carried out as a matrix multiplication with XNOR and popcounting ($\otimes$):

$$\mathrm{Linear}(\mathbf{A}) \approx \mathbf{Q_B}(\mathbf{W})\mathbf{Q_B}(\mathbf{A}) = \alpha(\mathbf{W_B} \otimes \mathbf{A_B}). \quad (1)$$

The simplest $\mathbf{Q_B}$ is the symbolic function, sign:

$$\mathbf{Q_B}(x) = \mathrm{sign}(x) = \begin{cases} -1, & \text{if } x < 0 \\ +1, & \text{if } x \geq 0 \end{cases}, \quad (2)$$

In backward propagation, gradient can't be directly calculated through the sign. Straight-through estimators (STE) are used to estimate gradient:

$$\frac{\partial \,\mathrm{sign}(x)}{\partial x} \approx \begin{cases} 1 & \text{if } |x| \leq 1 \\ 0 & \text{otherwise} \end{cases}. \quad (3)$$

Many works make efforts to find more effective binarization functions. BiTs binarize weights by $\mathbf{Q}_{\mathbf{B},w}$, and binarize activations by $\mathbf{Q}_{\mathbf{B},a}$ respectively:

$$\mathbf{Q}_{\mathbf{B},w}^{(-1,+1)}(\mathbf{W}) = \frac{\|\mathbf{W}\|_{l1}}{n_{\mathbf{W}}} \mathrm{sign}\left(\mathbf{W} - \overline{\mathbf{W}}\right), \quad (4)$$

$$\mathbf{Q}_{\mathbf{B},a}^{(-1,+1)}(\mathbf{A}) = \alpha \mathrm{sign}\left(\mathbf{A} - \beta\right), \quad (5)$$

where α, β are trainable parameters. Omitting scaling factors, both weights and activations have the same binarization level $\{-1, +1\}$. Different from BiTs, we remove the $\{0, 1\}$ binarization level in linear layers because different binarization levels need special transformation to avoid the ternary value problem. All linears are binarized as Eq.4, 5 in our general baseline.

Binary self-attention. FP self-attention is defined as cascaded matrix productions between the query, key and value:

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}. \quad (6)$$

Binary self-attention also consists of two steps: calculating self-attention map, $\mathbf{Att}$, and reweight value with binarized attention map respectively:

$$\mathbf{Att} \approx \mathrm{softmax}\left(\frac{\mathbf{Q}_{\mathbf{B},a}^{(-1,+1)}(\mathbf{Q})\mathbf{Q}_{\mathbf{B},a}^{(-1,+1)}(\mathbf{K^T})}{\sqrt{d_k}}\right), \quad (7)$$

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \approx \mathbf{Q}_{\mathbf{B},a}^{(0,+1)}(\mathbf{Att})\mathbf{Q}_{\mathbf{B},a}^{(-1,+1)}(\mathbf{V}), \quad (8)$$

where the binarization function for the attention map is defined as follows in BiTs:

$$\mathbf{Q}_{\mathbf{B},a}^{(0,+1)}(\mathbf{A}) = \alpha \left\lfloor \mathrm{Clip}\left(\frac{\mathbf{A} - \beta}{\alpha}, 0, 1\right)\right\rceil, \quad (9)$$

$$\lfloor \mathrm{Clip}\left(x, 0, 1\right)\rceil = \begin{cases} 0, & \text{if } x < 0.5 \\ 1, & \text{if } x \geq 0.5 \end{cases}.$$

After binarization, values in the attention map become $\{0, 1\}$ and formulate hard attention (omitting scaling factors). However, in Eq. 8, matrix production between $\mathbf{Att_B} \in \{0, +1\}^n$ and $\mathbf{V_B} \in \{-1, +1\}^n$ can't directly be implemented by XNOR and popcount at inference, which needs ternary operations in domain $\{-1, 0, +1\}$. It consumes double binary operations to transform ternary to binary operations:

$$\mathbf{Att}_{(\mathbf{0},\mathbf{1})}\mathbf{V_B} = (\mathbf{Att_B} \otimes \mathbf{V_B} + \mathbf{1} \otimes \mathbf{V_B}) >> 1, \quad (10)$$

where $\mathbf{Att_B}, \mathbf{V_B} \in \{-1, +1\}^n$, $>>$ is bitshift, and $\mathbf{Att_B}$ is constructed by directly replacing 0 as -1 in $\mathbf{Att}_{(\mathbf{0},\mathbf{1})}$.

Pretrain Binary Transformers

In this section, we propose pretrained foundation transformers based on the baseline architecture, termed BiPFT-A. We use simple but efficient pretraining tasks following the vanilla BERT and task-agnostic distillation (Wang et al. 2020). The pretraining tasks in BERT include masked language model and next sentence prediction. Additionally, inspired by the phenomenon that task-agnostic distillation improves pretraining efficiency in small models, we add distillation loss for both token and sentence-level features. In summary, the pretraining objectives of BiPFTs include:

method	BP	dist.	HS	binarization
BiT	✗	✓	✓	direct binarization
baseline*	✗	✓	✓	direct binarization
baseline	✗	✗	✗	direct binarization
BiPFT-A	✓	✗	✗	direct binarization
BiPFT-B	✓	✗	✗	error estimatation

Table 1: Summarization of proposed models, where BP indicates binary pretraining; dist. indicates task-specific distillation; HS indicates hyperparameter search in specific tasks.

Masked Language Model (ℓ_{MLM}): MLM objective is defined as minimizing the cross-entropy loss between the real and the prediction of masked tokens. Following BERTs, we randomly select 15% of the input tokens. Among these chosen tokens, 80% are swapped with [MASK], 10% are maintained as they are, and the remaining 10% tokens are substituted with a token randomly picked from the vocabulary.
Next Sentence Prediction (ℓ_{NSP}): NSP is defined as a binary classification task, where the objective is to predict if two segments appear consecutively in the source text. Following BERTs, we construct positive samples by selecting sequential sentences from the text corpus and negative samples are by pairing sentences from separate documents. The probability of positive and negative samples is equal.
Task-agnostic Distillation (ℓ_{logit}, ℓ_{rep}): previous works (Hinton, Vinyals, and Dean 2015) has shown minimizing KL divergency between model logits of the student and teacher achieves better performance than direct training. Following task-agnostic distillation (Sun et al. 2020; Wang et al. 2020), we distill logits in the last layer during pretraining. To improve convergency, we additionally apply L2 loss to distill hidden states layer by layer.
We use the aforementioned objectives to jointly train binary transformers in extensive pretraining data:

$$\ell_{\text{total}} = \ell_{\text{MLM}} + \ell_{\text{NSP}} + \frac{1}{n}\sum_{i=1}^{n} \ell_{\text{rep}}^{i} + \ell_{\text{logit}}\,. \tag{11}$$

After pre-training, task-agnostic knowledge significantly enhances the learning ability of the baseline models in various downstream tasks. As shown in Fig. 1 (b), once pretraining finished, we finetune the binary foundation model in various downstream tasks the same as full-precision cases, which bridges the training gap between full-precision and binary foundation models.

Estimate Binarization Polynomials

With the help of the pretraining phase, we explore how to better simulate self-attention with binary representations. To make full use of pretraining data, we investigate data-driven binarization methods. In this section, we first analyze binarization errors in self-attention and then propose binarization error estimators to achieve accurate binary self-attention. We add binarization error estimators to baseline architecture and pretrain this binary transformer as BiPFT-B. A summarization of proposed models is shown in Table 1.

Self-attention involves cascaded multiplications, making it challenging for previous empirical binarization designs:
Dynamic binary value. Previous BNNs focused on a more accurate simulation of matrix multiplication between real-valued weights and activations, where activations are dynamic values changing with input, and weights are fixed parameters. However, in self-attention, both items of matrix multiplication are dynamic values changing with inputs.
Cascaded multiplications. Self-attention has cascaded matrix multiplications. The error accumulation caused by direct binarization affects the accuracy of binary features. For instance, binarization errors from the matrix multiplication of keys and queries undoubtedly impact the following reweight between attention scores and values.

To find where binarization errors occur in self-attention, we first compare the differences before and after binarization; then define the residual polynomials ignored previously; and finally, we use low-rank estimators to model these residuals. In order to decompose the binarization error, we define binarization residuals of the query, key, and value items as well as their weights:

$$\begin{aligned} \mathbf{Q}^* \overset{\text{def}}{=} \mathbf{Q} - \mathbf{Q_B}, \quad \mathbf{K}^* \overset{\text{def}}{=} \mathbf{K} - \mathbf{K_B} \quad , \mathbf{V}^* \overset{\text{def}}{=} \mathbf{V} - \mathbf{V_B}, \\ \mathbf{W}^* = \mathbf{W} - \mathbf{W_B} \quad . \end{aligned} \tag{12}$$

According to Eq. 6, we first focus on the attention score between keys and queries. The full-precision $\mathbf{Q}$ and $\mathbf{K}$ can be decomposed into the sum of their binarized parts, $\mathbf{Q_B}$ and $\mathbf{K_B}$, and their binarization residuals, $\mathbf{Q}^*$ and $\mathbf{K}^*$. As shown in Eq. 13, in the simplified polynomials, the first term can be represented as directly binarized multiplication between $\mathbf{Q}$ and $\mathbf{K}$, while the other three terms constitute the quantization error:

$$\begin{aligned} \mathbf{A_{score}} &= \mathbf{Q}\mathbf{K^T} \\ &= (\mathbf{Q_B} + \mathbf{Q}^*)(\mathbf{K_B^T} + \mathbf{K}^{*\mathbf{T}}) \\ &= \mathbf{Q_B}\mathbf{K_B^T} + \underbrace{\mathbf{Q_B}\mathbf{K}^{*\mathbf{T}} + \mathbf{Q}^*\mathbf{K_B^T} + \mathbf{Q}^*\mathbf{K}^{*\mathbf{T}}}_{residual\ polynomials}. \end{aligned} \tag{13}$$

Previous binary operations are mainly designed for linear or convolutional layers in computer vision, with a lack of consideration for the multiplication between activations in self-attention. Directly replacing real matrix multiplication with binarized matrix multiplication after quantization overlooks the residual polynomials in Eq. 13, leading to binarization errors. Towards accurate binary self-attention, we propose data-driven estimators to model these binarization residual polynomials. We indicate residual polynomials in Eq. 13 as $\mathbf{A^*_{score}}$ and model these items by low-rank estimators:

$$\begin{aligned} \mathbf{A^*_{score}} &= \mathbf{A}\mathbf{W_q}\mathbf{W_k^{*T}}\mathbf{A^T} + \mathbf{A}\mathbf{W_q^*}\mathbf{W_k^T}\mathbf{A^T} \\ &+ \mathbf{A}\mathbf{W_q^*}\mathbf{W_k^{*T}}\mathbf{A^T} \\ &\approx \mathbf{A}\mathbf{w_q}\mathbf{w_k^{*T}}\mathbf{A^T} + \mathbf{A}\mathbf{w_q^*}\mathbf{w_k^T}\mathbf{A^T} \\ &+ \mathbf{A}\mathbf{w_q^*}\mathbf{w_k^{*T}}\mathbf{A^T}, \end{aligned} \tag{14}$$

where $\mathbf{W^{(*)}_{q,k}} \in \mathbf{R^{C\times C}}$, $\mathbf{w^{(*)}_{q,k}} \in \mathbf{R^{C\times 1}}$ and $\mathbf{C}$ donates hidden size of the transformer. In Eq. 14, $\mathbf{w^{(*)}_{q,k}}$ are train-

Quant	E-W-A	Size (MB)	FLOPs (G)	STS-B	MRPC	RTE	QQP	QNLI	SST-2	COLA	MNLI-m/mm	Avg.
BERT_{base}	32-32-32	418	22.5	88.8	86.5	66.8	91.4	91.3	92.9	55.8	83.4/83.6	82.3
Q-BERT	2-8-8	43.0	6.5	–	68.3	52.7	–	–	84.6	–	76.6/77.0	–
Q2BERT	2-8-8	43.0	6.5	4.4	68.4	52.7	67.0	61.3	80.6	0	47.2/47.3	47.7
TernaryBERT	2-2-8	28.0	6.4	–	87.5	68.2	90.1	–	–	50.7	83.3/83.3	–
BinaryBERT	1-1-8	16.5	3.1	88.6	85.5	72.2	91.2	91.5	92.6	53.4	84.2/84.7	82.7
BinaryBERT	1-1-4	16.5	1.5	87.2	83.3	65.3	91.2	90.9	92.3	44.4	83.9/84.2	79.9
Finetuning with task-specific distillation												
BinaryBERT	1-1-1	16.5	0.4	6.1	68.3	52.7	66.2	51.5	53.2	0	35.6/35.3	41.0
BiBERT	1-1-1	13.4	0.4	33.6	72.5	57.4	84.8	72.6	88.7	25.4	66.1/67.5	63.2
Baseline*	1-1-1	14.7	0.4	53.4	76.0	56.7	85.5	84.2	85.7	21.9	74.8/75.4	68.1
Finetuning without task-specific distillation												
Baseline	1-1-1	14.7	0.4	19.7	70.3	57.0	78.0	60.2	79.7	16.5	58.8/58.7	55.4
BiPFT-A	1-1-1	14.7	0.4	79.0	74.0	60.6	82.8	80.3	85.6	19.8	70.3/70.8	69.2
BiPFT-B	1-1-1	14.9	0.4	80.2	76.2	66.1	83.7	81.7	86.2	22.9	69.5/70.6	70.8
BinaryBERT	1-1-2	16.5	0.8	6.5	68.3	52.7	79.9	52.6	82.5	14.6	62.7/63.9	53.7
BiT	1-1-2	14.7	0.8	82.2	78.4	58.1	87.1	89.3	90.8	32.1	82.1/82.5	75.0
BiPFT-B	1-1-2	14.9	0.8	87.0	84.1	66.1	89.0	86.6	88.1	36.2	77.0/76.9	76.8

Table 2: Comparison of BERT quantization methods on the GLUE dev set. The E-W-A refers to the bit-width of embeddings, weights and activations. The baseline and baseline* are described in Table 1, which have almost the same architecture as BiT but evaluated in our common settings.

able parameters and used as approximations of $\mathbf{W}^{(*)}_{\mathbf{q},\mathbf{k}}$ respectively. In that case, the original dense matrix multiplications $\mathbf{A}\mathbf{W}^{(*)}_{\mathbf{q},\mathbf{k}}$ are approximated as low-rank multiplications $\mathbf{A}\mathbf{w}^{(*)}_{\mathbf{q},\mathbf{k}}$. We set the rank number as 1 to save 768× operations in the base-sized BERT, which will not introduce much additional cost. As a result, low-rank multiplications approximate residual polynomials ignored by direct binarization. In BiPFT-B, Eq. 7 in the baseline model is replaced by Eq. 15:

$$\mathbf{Att} \approx \text{softmax}\left(\frac{\mathbf{Q_B}\mathbf{K_B^T} + \mathbf{A^*_{score}}}{\sqrt{d_k}}\right). \tag{15}$$

We apply a similar analysis to the reweight multiplication in Eq. 8. We decompose full-precision value $\mathbf{V}$ into binary $\mathbf{V_B}$ and its resdual $\mathbf{V}^*$. Binarization residual polynomial can be represented as $\mathbf{Att}(\mathbf{A}\mathbf{W}^*_{\mathbf{v}})$:

$$\begin{aligned} \text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) &= \mathbf{AttV} \\ &= \mathbf{Att}(\mathbf{V_B} + \mathbf{V}^*) \\ &= \mathbf{AttV_B} + \mathbf{Att}(\mathbf{A}\mathbf{W}^*_{\mathbf{v}}). \end{aligned} \tag{16}$$

In contrast to decomposing attention map, $\mathbf{Att}$, we directly use binarized attention map, $\mathbf{Att_B}$, because binary attention map formulates hard attention that we don't want to break. In BiPFT-B, Eq. 8 in the baseline is replaced by Eq. 17:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) \approx \mathbf{Att_B}\mathbf{V_B} + \mathbf{Att_B}(\mathbf{A}\mathbf{w}^*_{\mathbf{v}}). \tag{17}$$

Experiments

Experiment Settings

In this work, we pursue aligning training settings between binary and full-precision (FP) transformers in both pretraining and finetuning phases, which is helpful to bridge the training gap between binary and FP transformers.

We keep the pretraining settings of BiPFTs similar to BERTs. In detail, we train the same architectured binary BERT models in the base size with 110M parameters. We quantize weights and embeddings in transformers to the bit-width of 1 and quantize activations to the bit-width of 1 and 2 respectively. In pretraining, we use the BooksCorpus (Zhu et al. 2015) and English Wikipedia (Devlin et al. 2018) as training data, including 800M and 2500M words respectively. The same as BERTs, lists, tables, and headers are ignored in Wikipedia. In preprocessing, we follow the BERT and use the WordPiece tokenizer (Devlin et al. 2018) with a 30522 vocabulary size. The max length of each sentence is set to 128 tokens. And the batch size is set to 512 in one step. There are total 5×10^5 steps in pretraining which include about 3 epochs of all data. The same as full-precision conditions, we train binary models with an AdamW optimizer with a 2×10^{-4} peak learning rate and 0.01 weight decay. A linear learning rate schedular with 5000 steps warm-up is also used. Our experiments show these common hyperparameters for most full-precision pretraining BERTs are general and robust enough for binary transformer pretraining.

In downstream tasks, we use the GLUE benchmark (Wang et al. 2018) to evaluate NLU performance. There are 8 subsets including CoLA, STS-B, MRPC, RTE, QQP, MNLI, QNLI. In finetuning, we also keep the same FP settings. In detail, we keep a constant 2×10^{-5} learning rate and 32 batchsize for all the subsets, and we keep the same training epochs as previous BiBERTs and BiTs. Notice that, we don't adapt to the best learning rate or batchsize for GLUE subsets like previous state-of-the-art works, which can improve performance a lot for BNNs but may result in overestimation of performance given new tasks.

Main Results

Table 2 shows comparisons with previous state-of-the-art BERTs in some low-bit quantization and binary. More detailed robustness and pretraining analysis are reported in Fig. 2, 3, 4 respectively.

Performance of BiPFT-A. We summarize methods in Table 1, where the baseline, baseline* and BiPFT-A share the same architecture; additional estimators are attached in BiPFT-B.

We use baseline* to implement the previous task-specific pipeline in **Fig. 1 (a)**. Baseline* uses the same hyperparame-

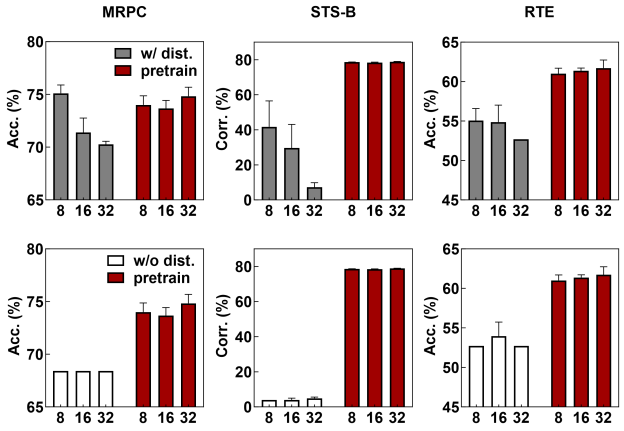


Figure 2: Comparisons of BiPFT-A and baselines in different batch sizes. Up: baseline with task-specific distillation; down: baseline without task-specific distillation.

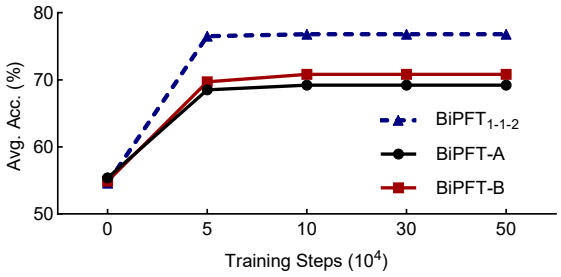


Figure 3: Pertraining performance in different training steps.

ters searched by BiTs and is trained with a distillation, while evaluated in the common settings. Baseline* is competitive to surpass BiBERT by 4.9% on average.

To evaluate binary BERT performance in general settings, we remove task-specific hyperparameter search and distillation. The baseline model withdraws 12.8% average accuracy dramatically, which indicates the weakness of binary BERT itself. This indicates the performance heavily relies on special training settings in task-specific binary BERTs.

After pretraining, BiPFT-A improves 13.9% compared with baseline; even if compared with baseline* with additional distillation and hyperparameter search, BiPFT-A surpasses 1.1%. This is the first time BNNs get rid of FP teachers and achieve better accuracy, which indicates pretraining significantly improves the learning ability of BNNs.

Performance of BiPFT-B. We report the performance of BiPFT-B in Table 2. With the estimation of binarization residual polynomials, BiPFT-B further improves 1.6% average performance compared with BiPFT-A. This indicates a large amount of pretraining data helps BNNs learn how to binarization in downstream. In total, the binary pretrained foundation model exceeds 15.4% average performance compared with baseline, which narrows 57.2% performance gap from the binary baseline to the FP BERT. In the setting of 2-bit activations, we also observe higger performance.

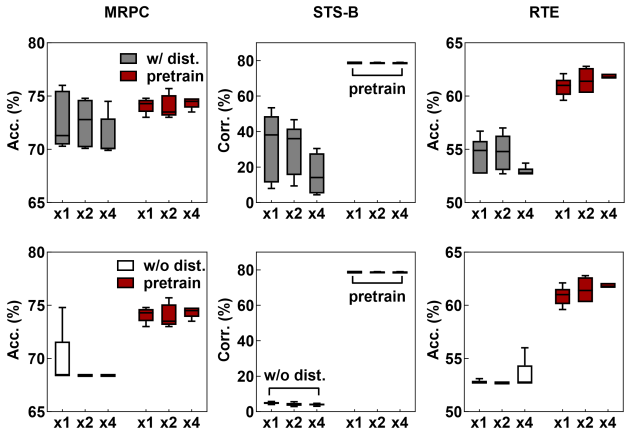


Figure 4: Comparisons of BiPFT-A and baselines in different learning rates. Up: baseline with task-specific distillation; down: baseline without task-specific distillation. We set the base learning rates for baselines according to searched results of BiTs for every task; we set learning rates for BiPFT-A from $\{5 \times 10^{-6}, 1 \times 10^{-5}, 2 \times 10^{-5}\}$.

In RTE, MRPC and STS-B datasets, there are 2.5k, 3.7k and 7k data respectively and they are relatively small datasets in GLUE. We observe BiPFT-B has more significant improvements than relatively big subsets, which are 9.1%, 5.9% and 60.5% in RTE, MRPC and STS-B. Even if compared with distilled baseline*, BiPFT-B surpasses 9.4%, 0.2% and 26.8% on average respectively. This indicates knowledge distillation is hard to make up the performance drop caused by the missing binary pretrained foundation models, in small downstream datasets.

Efficiency analysis. In Table 2, we compare operations and memory usage between FP and low-bit models. Compared with FP BERTs in base size, BiPFTs-B saves 56× operations and 28× memory for the 1-bit activations, while saves 28× operations and 28× memory for 2-bit activations.

Robustness analysis of binary transformers. We select three datasets, RTE, MRPC, STS-B and analyze robustness in different training settings. In Fig. 2, we compare BiPFT-A and baseline models in different batchsize settings to evaluate the robustness of pretraining in different batchsizes. In Fig. 4, we compare BiPFT-A and baselines in different learning rate settings to evaluate the robustness of pretraining in different learning rates. More detailed results are shown in Appendix B of our extended version (Xing et al. 2023).

Our observations are mainly in three aspects. Firstly, in Fig. 2 (up), compared with baseline* with distillation, BiPFT-A keeps almost higher performance stably in different batchsizes. Although task-specific distillation achieves at most 76.0% acc. in MRPC in batchsize 8, performance drops dramatically when training batchsize increases. The small batchsize makes it more challenging for parallel computation. Secondly, in Fig. 4 (up), we train baseline* and BiPFT-A in different learning rates and evaluate the variance of results. In different learning rates, pretraining helps to stabilize performance significantly. Because of unstable per-

Model	KQ↑	AttV↑	LoRA	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	$\text{MNLI}_{\text{-m/mm}}$	Avg.
FP-BERT_{base}	–	–	–	91.4	91.3	92.9	55.8	88.8	86.5	66.8	83.4/83.6	82.3
W/o pretraining												
Baseline	✗	✗	–	78.0	60.2	79.7	16.5	19.7	70.3	57.0	58.8/58.7	55.4
Baseline	✓	✓	–	78.2	60.6	79.5	11.1	19.5	70.3	55.6	59.5/59.7	54.9
Ablation in architectures												
BiPFT-A	✗	✗	–	82.8	80.3	85.6	19.8	79.0	74.0	60.6	70.3/70.8	69.2
BiPFT-A	✗	✓	–	84.0	81.5	85.9	19.4	79.4	73.8	63.2	70.5/71.0	69.9
BiPFT-A	✓	✗	–	83.7	81.7	85.7	21.3	79.3	76.2	62.8	70.0/70.9	70.2
BiPFT-B	✓	✓	–	83.7	81.7	86.2	22.9	80.2	76.2	66.1	69.5/70.6	70.8
Ablation in ranks												
BiPFT-B-rank1	✓	✓	–	83.7	81.7	86.2	22.9	80.2	76.2	66.1	69.5/70.6	70.8
BiPFT-B-rank2	✓	✓	–	83.3	80.9	87.3	18.7	78.8	73.5	65.3	69.4/70.4	69.7
BiPFT-B-rank4	✓	✓	–	83.4	81.6	86.2	18.9	76.9	75.0	61.0	69.7/70.4	69.2
Comparison with LoRA												
BiPFT-A	✗	✗	✓	83.5	81.4	85.3	18.1	80.7	76.5	62.8	70.1/70.9	69.9
BiPFT-B	✓	✓	✓	81.9	79.3	83.9	16.5	79.4	73.3	59.9	68.3/69.8	68.0
BiPFT-B	✓	✓	✗	83.7	81.7	86.2	22.9	80.2	76.2	66.1	69.5/70.6	70.8

Table 3: Ablation studies for BiPFTs. KQ↑ indicates adding estimators for key and query as Eq. 14; AttV↑ indicates adding estimators for value as Eq. 17.

formance, previous binary transformers have to perform hyperparameter search for different tasks, which can be inefficient and unstable. Thirdly, binary transformers heavily rely on distillation. When without pretraining, there are weak learning capabilities as shown in Fig 2 (down), 4 (down). In MRPC dataset, binary classification accuracy directly drops to 68.4% which is similar to encounter model degeneration or random choice; in STS-B dataset, the pearson and spearmanr close to 0%. These phenomena indicate task-specific binary transformers have a high risk to lost learning ability once removing FP teachers.

Pretraining time analysis. Fig. 3 shows the average GLUE performance of BiPFTs in different pretraining steps. In early pretraining time, downstream performance improves with more training steps. For the base-sized binary BERTs with 110M binary parameters, 1×10^5 pretraining steps are enough for fully pretraining, where the batch size is 512. This confirms enough training time for binary transformers.

Ablation Studies

Table 3 shows ablation studies for BiPFTs. More ablations for initialization are shown in Appendix C of our extended version (Xing et al. 2023).

Ablation in architectures. In BiPFT-B, we estimate binarization residual polynomials in two steps according to Eq. 14, 17 respectively. As shown in Table 3, with pretraining, it improves average performance when using estimators in Eq. 14 or Eq. 17 alone. When combining the estimations in both Eq. 14 and 17 together, it carries out the best and totally improves 1.6% performance on average. This confirms that low-rank multiplications have the capacity to learn to estimate binarization residual polynomials from queries, keys and values accordingly. However,data-driven binarization polynomial estimators are data-hungry. When without pretraining, estimators can't achieve better performance.

Ablation in ranks. We use the low-rank matrix multiplications as binarization polynomial estimators. By default, we use rank number 1 to reduce computational cost. To investigate the influence of ranks, We revise the rank number in Eq. 14 as 2 and 4. In Table 3, increasing the rank can't improve performance, which indicates larger ranks may encounter overfitting to binarization residual polynomials.

Comparison with LoRA. Because we use low-rank binarization estimators to improve binary multiplications between queries, keys and values, one potential idea could be whether we can use LoRA (Hu et al. 2022) to improve linear layers in self-attention. As shown in Table 3, although adding LoRA to binary transformers alone improves results, LoRA is less efficient compared with low-rank estimators of binarization residual polynomials. Moreover, when we both add LoRA and binarization polynomial estimators, it encounters unstable performance in our experiments, because of overfitting of low-rank parameters. As a result, low-rank estimators of binarization polynomials are more efficient and explicable for binary self-attention.

Conclusion

This work proposes the first binary pretrained foundation model for NLU tasks, promoting BNNs to the era of pretraining. This provides a lot of conveniences to finetune accurate, robust and training efficient binary transformers in downstream tasks. In the future, we think it would be meaningful to pretrain binary foundation models for natural language generation (NLG) tasks like current GPT (Brown et al. 2020) and LLama (Touvron et al. 2023), instead of downstream binary models. General knowledge is able to significantly improve the learning capabilities of BNNs.

Acknowledgments

This work is supported by the National Key R&D Program of China (No.2022ZD0116301) and the National Science Foundation of China under grant No.62206150. This work is also supported by NSFC No.62106249.

References

Bai, H.; Zhang, W.; Hou, L.; Shang, L.; Jin, J.; Jiang, X.; Liu, Q.; Lyu, M.; and King, I. 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)*, 4334–4348. Online: Association for Computational Linguistics.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Carlini, N.; Ippolito, D.; Jagielski, M.; Lee, K.; Tramer, F.; and Zhang, C. 2023. Quantifying Memorization Across Neural Language Models. In *The Eleventh International Conference on Learning Representations*.

Castano, A.; Alonso, J.; González, P.; and del Coz, J. J. 2023. An Equivalence Analysis of Binary Quantification Methods. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 6944–6952.

Chen, S.; Xie, E.; Ge, C.; Chen, R.; Liang, D.; and Luo, P. 2023. CycleMLP: A MLP-like Architecture for Dense Visual Predictions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Courbariaux, M.; Hubara, I.; Soudry, D.; El-Yaniv, R.; and Bengio, Y. 2016. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, Z.; Jiang, G.; Zhang, S.; Guo, L.; and Lin, W. 2023. SKDBERT: Compressing BERT via Stochastic Knowledge Distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 7414–7422.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2022. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

Gordon, M.; Duh, K.; and Andrews, N. 2020. Compressing BERT: Studying the Effects of Weight Pruning on Transfer Learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, 143–155.

He, B.; Martens, J.; Zhang, G.; Botev, A.; Brock, A.; Smith, S. L.; and Teh, Y. W. 2023. Deep Transformers without Shortcuts: Modifying Self-attention for Faithful Signal Propagation. In *The Eleventh International Conference on Learning Representations*.

He, P.; Gao, J.; and Chen, W. 2023. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. In *The Eleventh International Conference on Learning Representations*.

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

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Kim, S.; Gholami, A.; Yao, Z.; Mahoney, M. W.; and Keutzer, K. 2021. I-bert: Integer-only bert quantization. In *International conference on machine learning*, 5506–5518. PMLR.

Kirillov, A.; Mintun, E.; Ravi, N.; Mao, H.; Rolland, C.; Gustafson, L.; Xiao, T.; Whitehead, S.; Berg, A. C.; Lo, W.-Y.; et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643*.

Kunes, R. Z.; Yin, M.; Land, M.; Haviv, D.; Pe'er, D.; and Tavaré, S. 2023. Gradient Estimation for Binary Latent Variables via Gradient Variance Clipping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 8405–8412.

Li, Z.; Wang, Z.; Tan, M.; Nallapati, R.; Bhatia, P.; Arnold, A.; Xiang, B.; and Roth, D. 2022. Dq-bart: Efficient sequence-to-sequence model via joint distillation and quantization. *arXiv preprint arXiv:2203.11239*.

Lin, M.; Ji, R.; Xu, Z.; Zhang, B.; Chao, F.; Lin, C.-W.; and Shao, L. 2022. Siman: Sign-to-magnitude network binarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5): 6277–6288.

Liu, Z.; Oguz, B.; Pappu, A.; Shi, Y.; and Krishnamoorthi, R. 2023. Binary and Ternary Natural Language Generation. *arXiv preprint arXiv:2306.01841*.

Liu, Z.; Oguz, B.; Pappu, A.; Xiao, L.; Yih, S.; Li, M.; Krishnamoorthi, R.; and Mehdad, Y. 2022. Bit: Robustly binarized multi-distilled transformer. *Advances in neural information processing systems*, 35: 14303–14316.

Liu, Z.; Shen, Z.; Savvides, M.; and Cheng, K.-T. 2020. ReActNet: Towards Precise Binary Neural Network with Generalized Activation Functions. In *European Conference on Computer Vision (ECCV)*.

Liu, Z.; Wu, B.; Luo, W.; Yang, X.; Liu, W.; and Cheng, K.-T. 2018. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *Proceedings of the European conference on computer vision (ECCV)*, 722–737.

Li'an Zhuo, B. Z.; Chen, H.; Yang, L.; Chen, C.; Zhu, Y.; and Doermann, D. 2020. Cp-nas: Child-parent neural architecture search for 1-bit cnns. IJCAI.

Martinez, B.; Yang, J.; Bulat, A.; and Tzimiropoulos, G. 2020. Training binary neural networks with real-to-binary convolutions. *arXiv preprint arXiv:2003.11535*.

OpenAI. 2023. GPT-4 Technical Report. *arXiv preprint arXiv:2303.08774*.

Qin, H.; Ding, Y.; Zhang, M.; Yan, Q.; Liu, A.; Dang, Q.; Liu, Z.; and Liu, X. 2022. Bibert: Accurate fully binarized bert. *arXiv preprint arXiv:2203.06390*.

Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; and Zhou, D. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017a. Attention is all you need. *Advances in neural information processing systems*, 30.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017b. Attention Is All You Need. arXiv:1706.03762.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; and Zhou, M. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33: 5776–5788.

Wang, X.; Zhang, X.; Cao, Y.; Wang, W.; Shen, C.; and Huang, T. 2023. Seggpt: Segmenting everything in context. *arXiv preprint arXiv:2304.03284*.

Xiao, G.; Lin, J.; Seznec, M.; Wu, H.; Demouth, J.; and Han, S. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, 38087–38099. PMLR.

Xing, X.; Du, L.; Wang, X.; Zeng, X.; Wang, Y.; Zhang, Z.; and Zhang, J. 2023. BiPFT: Binary Pre-trained Foundation Transformer with Low-rank Estimation of Binarization Residual Polynomials. arXiv:2312.08937.

Xing, X.; Jiang, Y.; Zhang, B.; Ding, W.; Li, Y.; Li, H.; and Peng, H. 2022a. Binary Dense Predictors for Human Pose Estimation Based on Dynamic Thresholds and Filtering. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1705–1709.

Xing, X.; Li, Y.; Li, W.; Ding, W.; Jiang, Y.; Wang, Y.; Shao, J.; Liu, C.; and Liu, X. 2022b. Towards accurate binary neural networks via modeling contextual dependencies. In *European Conference on Computer Vision*, 536–552. Springer.

Xu, S.; Li, Y.; Ma, T.; Lin, M.; Dong, H.; Zhang, B.; Gao, P.; and Lu, J. 2023. Resilient Binary Neural Network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 10620–10628.

Zhao, Q.; and Wressnegger, C. 2023. Holistic Adversarially Robust Pruning. In *The Eleventh International Conference on Learning Representations*.

Zhu, Y.; Kiros, R.; Zemel, R.; Salakhutdinov, R.; Urtasun, R.; Torralba, A.; and Fidler, S. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, 19–27.