Lifting the Curse of Capacity Gap in Distilling Language Models

Chen Zhang[♣], **Yang Yang**[♦], **Jiahao Liu**[♦], **Jingang Wang**[♦], **Yunsen Xian**[♦], **Benyou Wang**[♥], **Dawei Song**[♣*]
[♣]Beijing Institute of Technology [♦]Meituan NLP
[♥]The Chinese University of Hong Kong, Shenzhen
chenzhang9702@outlook.com

Abstract

Pretrained language models (LMs) have shown compelling performance on various downstream tasks, but unfortunately they require a tremendous amount of inference compute. Knowledge distillation finds a path to compress LMs to small ones with a teacher-student paradigm. However, when the capacity gap between the teacher and the student is large, a curse of capacity gap appears, invoking a deficiency in distilling LMs. While a few studies have been carried out to fill the gap, the curse is not yet well tackled. In this paper, we aim at lifting the curse of capacity gap via enlarging the capacity of the student without notably increasing the inference compute. Largely motivated by sparse activation regime of mixture of experts (MoE), we propose a mixture of minimal experts (MiniMoE), which imposes extra parameters to the student but introduces almost no additional inference compute. Experimental results on GLUE and CoNLL demonstrate the curse of capacity gap is lifted by the magic of MiniMoE to a large extent. MiniMoE also achieves the state-of-the-art performance at small FLOPs compared with a range of competitive baselines. With a compression rate as much as ~50×, MiniMoE preserves ~95% GLUE score of the teacher.[1]

1 Introduction

Pretrained language models (LMs) have become a popular choice for various downstream tasks, e.g., text classification, token classification, and question answering (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020). Unfortunately, appealing performance comes with a huge cost of inference compute due to the scale of LMs. Knowledge distillation (Hinton et al., 2015; Sun et al., 2019), as an alternative to model pruning (Han et al., 2015) and quantization (Sung et al., 2015), discovers a way to compress (Bucila et al., 2006) LMs with a teacher-student paradigm.

Table 1: The *curse of the capacity gap* in terms of GLUE (Wang et al., 2019). The △ denotes the performance difference of preceding two numbers. To ensure students at similar scales, the student/teacher scale ratios are properly reduced for some methods.

Method	BERT_base	BERT_large	△
Teacher	86.7	88.3	+1.6
$\text{KD}_{10\%/5\%}$ (2015)	81.3	80.8	−0.5
$\text{DynaBERT}_{15\%/5\%}$ (2020)	81.1	79.2	−1.9
$\text{MiniDisc}_{10\%/5\%}$ (2022a)	82.4	82.1	−0.3
$\text{TinyBERT}_{4L;312H}$ (2020)	82.7	82.5	−0.2
$\text{MiniLM}_{3L;384H}$ (2021b)	82.5	82.0	−0.5
$\text{MiniMoE}_{3L;384H}$ (**ours**)	82.6	83.1	+0.5

However, in LM distillation, we recognize a ***curse** of capacity gap* as:

"Large teachers, poor students."

The curse of capacity gap refers to a deficiency that a larger teacher might unexpectedly result in a poorer student especially when the capacity gap between the teacher and the student is large (Mirzadeh et al., 2020; Cho and Hariharan, 2019), as illustrated in Table 1. Notably, this is the first verification in LM distillation since previous studies recognize the curse in vision model distillation. Although a few studies (Wang et al., 2020; Zhang et al., 2022a; Park et al., 2021a) have investigated to fill the gap, the curse is still not yet tackled.

To the demand, we aim at lifting the curse of capacity gap via enlarging the capacity of the student without notably increasing the inference compute. We propose a mixture of minimal experts (MiniMoE), inspired by the intuition of sparse activation of mixture of experts (MoE) (Shazeer et al., 2017). Thanks to that the activation process can be parallel on either single or multiple devices (He et al., 2021; Rajbhandari et al., 2022), MiniMoE on the one hand imposes extra parameters to the

[*]Dawei Song is the corresponding author.
[1]Code is available at https://github.com/GeneZC/MiniMoE

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 4535–4553
July 9-14, 2023 ©2023 Association for Computational Linguistics

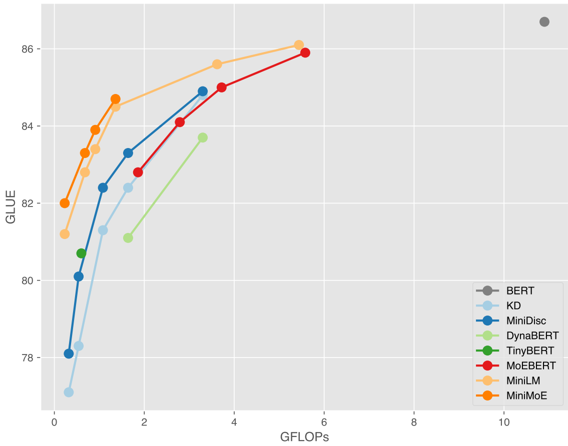

Figure 1: GLUE v.s. GFLOPs.

student, but on the other hand introduces negligibly additional inference compute brought by routing algorithm. To our best knowledge, this is the first work aiming at lifting the curse completely.

Experiments are conducted on GLUE (Wang et al., 2019) and CoNLL (Sang and Meulder, 2003). The results exhibit that MINIMOE largely lifts the curse of the gap as in Table 1. MINIMOE also achieves state-of-the-art performance compared with a range of competitive baselines, as shown in Figure 1. With compression as much as $\sim$50$\times$, MINIMOE preserves $\times$95% GLUE score of the teacher. Thereby, we state that MINIMOE is *a small yet nontrivial magic, making a great difference in lifting the curse*.

2 Curse of Capacity Gap

The curse of capacity gap is not new but is already recognized in studies on vision model distillation (Mirzadeh et al., 2020; Cho and Hariharan, 2019). While a hit-the-mind drawback of the curse is that the performance of distilling to a small student can be dramatically worse than that of distilling to a slightly larger one, a rather counter-intuitive deficiency is invoked as that the performance of distilling from a large teacher can be unexpectedly worse than that of distilling from a smaller one (i.e., *large teacher, poor student*). We here give a minor theoretical justification on the curse, as a plus to the empirical justification.

Proposition 1 (VC dimension theory, Vapnik, 1998)**.** *Assuming that the teacher function is $f_{\mathcal{T}} \in \mathcal{F}_{\mathcal{T}}$, the labeling function is $f \in \mathcal{F}$, and the data is $\mathcal{D}$, we have:*

$$r(f_{\mathcal{T}}) - r(f) \leq \epsilon_{\mathcal{T}} + o(\frac{|\mathcal{F}_{\mathcal{T}}|_c}{|\mathcal{D}|}),$$

where $r(\cdot)$ is the risk function, $|\cdot|_c$ is the function class capacity measure, and $|\cdot|$ is the data scale measure. It should be highlighted that the approximation error $\epsilon_{\mathcal{T}}$ is negatively correlated with the capacity of the teacher model while the estimation error $o(\cdot)$ is correlated with the learning optimization.

Proposition 2 (Generalized distillation theory, Lopez-Paz et al., 2016)**.** *Additionally providing that the student function is $f_{\mathcal{S}} \in \mathcal{F}_{\mathcal{S}}$, we have:*

$$r(f_{\mathcal{S}}) - r(f_{\mathcal{T}}) \leq \epsilon_{\mathcal{G}} + o(\frac{|\mathcal{F}_{\mathcal{S}}|_c}{|\mathcal{D}|^{\alpha}}),$$

where the approximation error $\epsilon_{\mathcal{G}}$ is positively correlated with the capacity gap between the teacher and the student models, and $1/2 \leq \alpha \leq 1$ is a factor correlated to the learning rate.

Theorem 1. *The bound for the student function at a learning rate can be written as:*

$$\begin{aligned} r(f_{\mathcal{S}}) - r(f) &\leq \epsilon_{\mathcal{T}} + \epsilon_{\mathcal{G}} + o(\frac{|\mathcal{F}_{\mathcal{T}}|_c}{|\mathcal{D}|}) + o(\frac{|\mathcal{F}_{\mathcal{S}}|_c}{|\mathcal{D}|^{\alpha}}) \\ &\leq \epsilon_{\mathcal{T}} + \epsilon_{\mathcal{G}} + o(\frac{|\mathcal{F}_{\mathcal{T}}|_c + |\mathcal{F}_{\mathcal{S}}|_c}{|\mathcal{D}|^{\alpha}}), \end{aligned}$$

Proof. The proof is rather straightforward by combining Proposition 1 and 2. □

Remark 1. *Under the same distillation setting, we can ignore the estimation error. When we compare two students of different capacities distilled from a teacher of the same capacity, the student of a smaller capacity has a larger $\epsilon_{\mathcal{G}}$ thus lower performance. When we compare two students of the same capacities distilled from teachers of different capacities, the student distilled from the teacher of a larger capacity has a smaller $\epsilon_{\mathcal{T}}$ yet a larger $\epsilon_{\mathcal{G}}$ thus a tradeoff.*

Remark 1 basically tells that a tradeoff is associated with the increase of teacher capacity, implying that increasing teacher capacity would first lead to improved but then degraded student performance. This tradeoff naturally corresponds with the curse.

On the other hand, it is accepted that large capacity gap is a pain and is processed in literature of LM distillation (Wang et al., 2020; Zhang et al., 2022a; Zhou et al., 2022). Being unaware of the curse of capacity gap, these studies attempt to offer student-friendly teachers by either interpolating teacher assistants (Wang et al., 2020; Zhang et al., 2022a) or adapting teacher knowledge (Zhou et al., 2022; Yang et al., 2022). The unawareness is largely due to a fun fact that they only distil LMs like $\text{BERT}_{\text{base}}$, but neglect the scalability to LMs like $\text{BERT}_{\text{large}}$ especially when the student is small. Though the performance of student can be boosted in this way, the curse still remains in LM distillation as in Figure 2. Other related work in knowledge distillation is given in Appendix E.

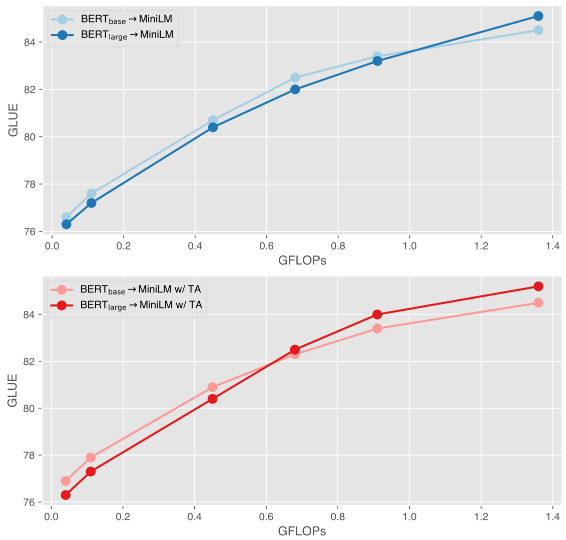

Figure 2: The performance of MiniLM and MiniLM w/ TA across different student scales upon distilling $\text{BERT}_{\text{base}}$. We are glad to share checkpoints of an array of scales, together with those of MINIMOE, to facilitate the development of related research. It should be noted the unit of a vertical grid is comparably large.

Embarrassingly, while the curse is claimed to be tackled in vision model distillation (Zhu and Wang, 2021; Park et al., 2021a; Zhao et al., 2022), our preliminary study (cf. Table 13 in Appendix H) indicates they are either expensive or not capable of LMs. The potential differences are as follows: tasks (e.g., ImageNet v.s. GLUE), backbones (e.g., ResNets v.s. transformers), and paradigms (e.g., from scratch v.s. pretraining).

3 MiniMoE

3.1 Motivation

Enlarging the capacity of the student is an intuitive solution to lift the curse of capacity gap. However, regarding the inference compute efficiency, the increase of capacity should not introduce much inference compute.

An initial proposal can be using quantized backbones (Zafrir et al., 2019; Bai et al., 2021). Quantized backbones may decrease the compute precision, therefore maintaining inference compute constant, along the course of enlarging the capacity. But a vital portion of hardware-specific modifications are needed to do so. We hence move on to next possibility.

Table 2: A comparison between MINIMOE and other possible alternatives.

Method	Flexible Hardware	Controllable Compute	Scalable Compute
Quantization	✗	✓	✓
Depth-adaptation	✓	✗	✓
MoEfication	✓	✓	✗
MINIMOE	✓	✓	✓

Another alternative is using dynamic networks (Han et al., 2021) based on the idea of conditional computation (Bengio et al., 2015). MoE computation (Shazeer et al., 2017; Fedus et al., 2021) is an option derived upon the sparse activation property to increase the scale with only minor losses in compute efficiency. The other commonly used one is depth-adaptive computation (Xin et al., 2020; Zhou et al., 2020; Goyal et al., 2020; Kim and Cho, 2021) which involves layers into computation adaptively on either example (alias early exiting, Xin et al., 2020; Zhou et al., 2020) or token (alias token reduction, Goyal et al., 2020; Kim and Cho, 2021) level. A critical distinction between MoE and depth-adaptive models is that the compute of an MoE model is accurately under control while that of a depth-adaptive model is not. We are impelled by the merits of MoE, and propose a MINIMOE so that the capacity of the student can be enlarged without much inference overhead increment.

Additionally, we argue that MINIMOE is orthogonal to alternatives mentioned above, and MINIMOE can be incorporated to these alternatives and makes it possible to serve more extreme scenarios. It is noteworthy that a certain stream of work (Zhang et al., 2022b; Zuo et al., 2022) actually accelerates LMs via precisely converting them into MoE models. Nonetheless, the moefication process is directly exerted to LMs with limited inference compute improvements (cf. MoEBERT in Figure 1). Contrarily, MINIMOE is comprised of minimal experts, each of which can be extremely small. A comparison between mentioned possibilities and MINIMOE is listed in Table 2. And other related work of interest is given in Appendix E.

3.2 Implementation

Minimal Language Models Typical language models are comprised of a stack of transformers layers (Vaswani et al., 2017), and are pretrained with language modeling tasks such as masked language modeling (Devlin et al., 2019). A transformer layer can be decomposed to a multi-head self-attention (MHA) block and a feed-forward network (FFN) block. Concretely, given an n-length sequence of d-dimension input vectors $\mathbf{X} \in \mathbb{R}^{n\times d}$ with the i-th vector being $\mathbf{x}_i$, the output of the MHA block with A independent heads can be represented as:

$$\mathrm{MHA}(\mathbf{X}) = \sum_{j=1}^{A} \mathrm{Attn}(\mathbf{X}; \mathbf{W}_j^{\mathsf{Q}}, \mathbf{W}_j^{\mathsf{K}})\mathbf{X}\mathbf{W}_j^{\mathsf{V}}\mathbf{W}_j^{\mathsf{O}},$$

$$\mathrm{Attn}(\mathbf{X};\mathbf{W}_j^{\mathsf{Q}}, \mathbf{W}_j^{\mathsf{K}}) = \mathrm{softmax}(\mathbf{X}\mathbf{W}_j^{\mathsf{Q}}\mathbf{W}_j^{\mathsf{K}\top}\mathbf{X}^\top/d^{\mathsf{A}}),$$

where the j-th head is parameterized by $\mathbf{W}_j^{\mathsf{Q}}$, $\mathbf{W}_j^{\mathsf{K}}$, $\mathbf{W}_j^{\mathsf{V}} \in \mathbb{R}^{d\times d^{\mathsf{A}}}$, and $\mathbf{W}_j^{\mathsf{O}} \in \mathbb{R}^{d^{\mathsf{A}}\times d}$. On the other hand, the output of the FFN block is shown as:

$$\mathrm{FFN}(\mathbf{X}) = \mathrm{GELU}(\mathbf{X}\mathbf{W}^{\mathsf{I}})\mathbf{W}^{\mathsf{O}},$$

where two fully-connected layers are parameterized by $\mathbf{W}^{\mathsf{I}} \in \mathbb{R}^{d\times d^{\mathsf{I}}}$ and $\mathbf{W}^{\mathsf{O}} \in \mathbb{R}^{d^{\mathsf{I}}\times d}$ respectively. Details like biases, normalizations of a transformer layer are omitted for brevity.

To reach an acceptable compute budget, pioneering studies either pretrain language models or distil ones of small scales from LMs as in Figure 3. There are three lines of work in LM distillation: firstly, task-specific distillation (Sun et al., 2019; Li et al., 2020; Sun et al., 2020a; Park et al., 2021b; Hou et al., 2020; Xia et al., 2022) that conducts distillation on a specific task at finetuning stage; secondly, task-agnostic distillation (Turc et al., 2019; Sanh et al., 2019; Sun et al., 2020b; Wang et al., 2021b) that conducts distillation at pretraining stage; and thirdly, two-stage distillation (Jiao et al., 2020) that combines the power of both task-agnostic and -specific distillation. Here, the distilled language models only refer to language models distilled with task-agnostic distillation regarding better task-scalability as the number of concerned tasks explodes.

We formally define the distilled language models as minimal language models (MiniLMs, somehow abuse of notation with Wang et al., 2020) notated with $\mathcal{S}$. In contrast, LMs are notated with $\mathcal{T}$. The

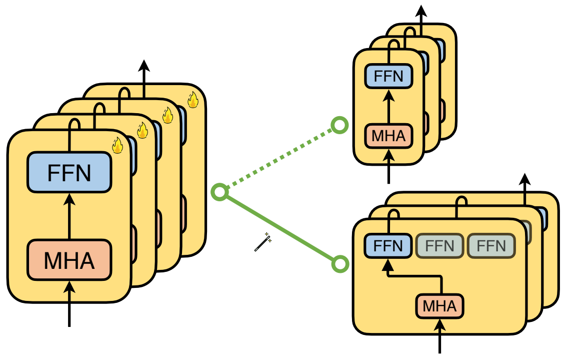

Figure 3: Implementation of MINIMOE.

learning objective of MiniLMs can be abstracted as $\mathcal{L}(\mathcal{S};\mathcal{T},\mathcal{D})$, where $\mathcal{D}$ denotes the data. The specific form of $\mathcal{L}$ can be adapted to arbitrary alignment strategies. We adopt a relation alignment strategy (Wang et al., 2021b) as follows:

$$\begin{aligned}\mathcal{L}(\mathcal{S};\mathcal{T},\mathcal{D}) &= \mathbb{E}_{\mathbf{X}\sim\mathcal{D}}\sum_{j=1}^{R} \\ &\mathrm{KL}(\mathrm{Reln}(\mathbf{X};{}^{\mathcal{T}}\mathbf{W}_j^{\mathsf{Q}}), \mathrm{Reln}(\mathbf{X};{}^{\mathcal{S}}\mathbf{W}_j^{\mathsf{Q}})) \\ &+ \mathrm{KL}(\mathrm{Reln}(\mathbf{X};{}^{\mathcal{T}}\mathbf{W}_j^{\mathsf{K}}), \mathrm{Reln}(\mathbf{X};{}^{\mathcal{S}}\mathbf{W}_j^{\mathsf{K}})) \\ &+ \mathrm{KL}(\mathrm{Reln}(\mathbf{X};{}^{\mathcal{T}}\mathbf{W}_j^{\mathsf{V}}), \mathrm{Reln}(\mathbf{X};{}^{\mathcal{S}}\mathbf{W}_j^{\mathsf{V}})),\end{aligned}$$

$$\begin{aligned}&\mathrm{Reln}(\mathbf{X};{}^{\mathcal{T}}\mathbf{W}_j^{\mathsf{Q}}) \\ &\quad = \mathrm{softmax}(\mathbf{X}^{\mathcal{T}}\mathbf{W}_j^{\mathsf{Q}\,\mathcal{T}}\mathbf{W}_j^{\mathsf{Q}\top}\mathbf{X}^\top/d^{\mathsf{R}}),\end{aligned}$$

where KL stands for kullback-leibler divergence. Essentially, relation heads are derived by merging the original A attention heads and then splitting them to R heads. ${}^{\mathcal{T}}\mathbf{W}_j^{\mathsf{Q}}$ is the redistributed query parameter of the j-th relation head within totally R heads from the last layer of the LM, likewise ${}^{\mathcal{T}}\mathbf{W}_j^{\mathsf{K}}$ and ${}^{\mathcal{T}}\mathbf{W}_j^{\mathsf{V}}$ are the key and value parameters. An auxiliary MHA block is employed as the last layer of the MiniLM for better alignment following Wang et al. (2021a). The MiniLM can be then finetuned on any tasks.

Mixture of Minimal Experts Naturally, in order to enlarge the learning capacity gap of the student, we should add more parameters to the student. However, trivially adding parameters usually leads to a loss of inference compute efficiency.

To remedy this, a mixture of minimal experts is proposed as in Figure 3. Following prior literature (Shazeer et al., 2017, 2018), if we consider a FFN block in a MiniLM as a minimal expert, then extra parameters are exactly imposed as minimal experts to be added to the FFN block. The FFN block is enabled as a mixture of m minimal experts FFN^{MoE} in an expert gating tactic as:

$$\text{FFN}^{\text{MoE}}(\mathbf{x}_i) = p_k(\mathbf{x}_i) \cdot \text{FFN}_k(\mathbf{x}_i),$$

$$p_k(\mathbf{x}_i) = \frac{\exp(\mathbf{x}_i \mathbf{w}_k^{\mathsf{G}})}{\sum_{j=1}^{m} \exp(\mathbf{x}_i \mathbf{w}_j^{\mathsf{G}})},$$

$$k = \text{argmax } p(\mathbf{x}_i),$$

where the j-th gate is parameterized by $\mathbf{w}_j^{\mathsf{G}} \in \mathbb{R}^d$, and correspondingly the j-th minimal expert is denoted as FFN_j. We further follow Fedus et al. (2021) to only allow top-*one* gating (i.e., only the expert with highest gating probability is reserved) because we want to keep the inference compute untouched. There are also diverse designs to achieve the sparse routing, such as hashing (Roller et al., 2021) which we find performs worse (cf. Figure 5).

Since only one minimal expert is activated during the inference, the compute is only negligibly increased by expert routing. As a complement, we can also achieve, if necessary, a mixture of experts in an MHA block similarly.

To encourage a balanced load across minimal experts, a differentiable load balancing objective $\mathcal{B}(\mathcal{S};\mathcal{D})$ is added from Lepikhin et al. (2021) as:

$$\mathcal{B}(\mathcal{S};\mathcal{D}) = \alpha \cdot m \sum_{j=1}^{m} f_j \cdot P_j,$$

$$f_j = \mathbb{E}_{\mathbf{x}_i \sim \mathcal{D}}[\mathbb{I}\{\text{argmax } p(\mathbf{x}_i), j\}],$$

$$P_j = \mathbb{E}_{\mathbf{x}_i \sim \mathcal{D}}[p_j(\mathbf{x}_i)],$$

where α is a coefficient that should be manually tune and is kept as 0.01 throughout this work following (Fedus et al., 2021). While f_j depicts the fraction of tokens dispatched to the j-th minimal expert, P_j describes the fraction of the routing probability to the j-th minimal expert. And a multiplier m is used to make the magnitude of the objective invariant to the number of minimal experts. The load balancing objective basically desires a uniform routing so that the loss can be minimized. The objective is added to the MiniLM not only at task-agnostic distillation stage but also at finetuning stage for practical concerns (cf. Figure 5).

4 Experiments

4.1 Data and Metrics

We conduct experiments on GLUE (Wang et al., 2019) and CoNLL (Sang and Meulder, 2003). The GLUE originally consists of two sequence classification tasks, SST-2 (Socher et al., 2013), i.e., CoLA (Warstadt et al., 2019), and seven sequence-pair classification tasks, i.e., MRPC (Dolan and Brockett, 2005), STS-B (Cer et al., 2017), QQP, MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), RTE (Bentivogli et al., 2011), WNLI (Levesque et al., 2012). We exclude WNLI and CoLA due to the evaluation inconsistency (in other words, MiniLMs get dramatically worse results while LMs get much better ones as found out in Xia et al., 2022) and use the left tasks. The CoNLL is a token classification task. Following BERT (Devlin et al., 2019), we report Accuracy (Acc) on SST-2, MNLI, QNLI, RTE, Spearman Correlation scores (SpCorr) on STS-B, and F1 on MRPC, QQP, CoNLL. Average score over tasks from GLUE (GLUE Score) is additionally computed. Results on development sets are reported. GFLOPs are also attached as theoretical speedup references. We adopt Wikipedia data for task-agnostic disitllation. The detailed statistics, maximum sequence lengths, and metrics of GLUE, CoNLL, and Wikipedia are supplied in Appendix A.

4.2 Hands-on Details

Experiments are conducted upon distilling $\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{large}}$ (Devlin et al., 2019). The distillation carried out on eight Nvidia A100s. The number of relation heads is set to 32. After the distillation, finetuning is carried out on one Nvidia A100. The number of minimal experts m is default to 4 otherwise specified. Other details are supplied in Appendix B. All experiments are task-agnostic ones, except those in Table 13.

4.3 Baselines

We compare MINIMOE with several state-of-the-art baselines.

Conventional Distillation FT indicates direct finetuning the student. KD (Hinton et al., 2015), PKD (Sun et al., 2019), and CKD (Park et al., 2021b) are methods with different distillation objectives, i.e., KD directly distills logits, PKD distills both logits and hidden states, and CKD distills high-order relations. While above four methods originally initialize student structures by dropping layers, we enable them with a global pruning so that they can adapt to students of small scales. DynaBERT (Hou et al., 2020) uses a two-step pruning

Table 3: The results of comparison between distilling BERT$_{\text{base}}$ and BERT$_{\text{large}}$.

Method	Teacher	SST-2 Acc	MRPC F1	STS-B SpCorr	QQP F1	MNLI-m/mm Acc	QNLI Acc	RTE Acc	GLUE Score	CoNLL F1
MiniLM$_{6L;384H}$	BERT$_{\text{base}}$	91.1	90.1	88.1	86.7	81.5/81.8	89.2	67.9	84.5	93.2
	BERT$_{\text{large}}$$\Uparrow$	90.9	90.6	89.0	86.9	81.8/82.4	88.8	70.0	85.1	93.2
w/ TA	BERT$_{\text{base}}$	91.3	90.3	88.2	86.8	81.4/81.6	89.7	66.8	84.5	93.2
	BERT$_{\text{large}}$$\Uparrow$	91.4	89.8	88.5	87.0	81.9/81.6	89.5	71.5	85.2	93.2
MiniMoE$_{6L;384H}$	BERT$_{\text{base}}$	91.3	90.2	88.6	86.5	81.6/81.5	89.5	68.6	84.7	93.3
	BERT$_{\text{large}}$$\Uparrow$[1]	90.5	90.0	88.8	86.8	81.8/82.2	90.8	70.4	85.2	93.3
MiniLM$_{4L;384H}$	BERT$_{\text{base}}$	90.0	88.6	87.2	86.1	80.0/80.3	87.9	67.2	83.4	91.5
	BERT$_{\text{large}}$$\Downarrow$	89.3	87.5	88.1	85.9	79.9/80.2	87.6	67.2	83.2	91.2
w/ TA	BERT$_{\text{base}}$	90.0	88.5	87.3	86.3	80.1/80.7	88.0	66.4	83.4	91.8
	BERT$_{\text{large}}$$\Uparrow$	90.6	88.7	88.1	86.3	80.5/80.7	87.9	69.0	84.0	92.2
MiniMoE$_{4L;384H}$	BERT$_{\text{base}}$	90.8	88.1	88.2	85.9	79.8/80.4	88.6	69.3	83.9	92.3
	BERT$_{\text{large}}$$\Uparrow$	90.5	88.0	88.7	86.7	80.9/80.9	89.2	69.0	84.2	92.4
MiniLM$_{3L;384H}$	BERT$_{\text{base}}$	89.1	89.1	86.6	85.4	77.8/78.4	87.2	66.1	82.5	90.1
	BERT$_{\text{large}}$$\Downarrow$	89.1	86.1	87.1	85.1	78.6/78.5	86.0	65.7	82.0	87.3
w/ TA	BERT$_{\text{base}}$	89.8	87.8	86.0	85.5	77.6/78.5	86.8	66.1	82.3	90.4
	BERT$_{\text{large}}$$\Downarrow$	89.7	84.9	87.2	85.2	78.5/79.1	86.6	66.4	82.2	90.2
MiniMoE$_{3L;384H}$	BERT$_{\text{base}}$	89.3	87.4	87.8	85.6	78.2/78.7	87.2	67.0	82.6	90.7
	BERT$_{\text{large}}$$\Uparrow$	89.1	88.4	87.6	86.2	78.8/79.5	87.5	67.9	83.1	91.6

[1] $\Uparrow$ is used to indicate the deficiency is tackled on both GLUE and CoNLL, otherwise $\Downarrow$ is used.

to regulate student structures and a distillation objective akin to PKD. MoEBERT (Zuo et al., 2022) moefies LMs by decomposing FFN blocks to MoE layers. For these task-specific distillation methods, student structures are denoted either with $_{*\text{L}}$ for preserved number of layers in layer-dropping or with $_{*\%}$ for preserved portion of parameters in pruning.

As aforementioned methods are task-specific distillation ones, we then introduce task-agnostic ones. TinyBERT (Jiao et al., 2020) exploits a distillation objective distilled with a combination of various feature alignments. MiniLM (Wang et al., 2021b) straightforwardly utilizes a distillation objective with a deep relation alignment exactly the same with ours. Since task-agnostic distillation allows both dropping layers and hidden dimensions, student structures are denoted with $_{*\text{L};*\text{H}}$ accordingly.

Capacity-aware Distillation MiniLM w/ TA (Wang et al., 2020) specifically incorporates a teacher assistant to MiniLM. MiniDisc (Zhang et al., 2022a) argues that the scale of the teacher assistant is crucial for student performance and proposes an automatic teacher assistant scheduler based on properties of pruning. While MiniLM w/ TA is only inspected under a task-agnostic setting, MiniDisc offers results under both task-specific and task-agnostic settings. Nevertheless, only task-specific MiniDisc is selected since pruned MiniLMs can be unfair to compare with. There is scarce work in this direction in which we find these two are the most comparable ones.

4.4 Main Results

From results in Table 3, we find that MiniMoE lifts the curse of capacity gap at all concerned times of compression. For example, MiniMoE$_{3L;384H}$ disitlled from BERT$_{\text{large}}$ has an absolute 0.5 performance gain over that distilled from BERT$_{\text{base}}$ on GLUE, and the value on CoNLL is 0.9. On another note, MiniLM is free of the curse only at small times of compression, and MiniLM w/ TA can somewhat saves MiniLM from the curse at intermediate times of compression. For example, both MiniLM$_{3L;384H}$ and MiniLM$_{3L;384H}$ w/ TA fail to improve the performance via replacing BERT$_{\text{base}}$ with BERT$_{\text{large}}$. Results on larger LMs like BERT$_{\text{xlarge}}$ are supplied in Appendix F for scalability check.

From results in Table 4, we also observe that MiniMoE generally outperforms both conventional and capacity-aware baselines and achieves new state-of-the-art performance at all concerned times of compression. For example, MiniMoE$_{4L;192H}$ has an absolute 0.8 performance improvement over MiniLM$_{4L;192H}$ on GLUE. And the reason why MiniMoE$_{3L;384H}$ slightly under-

Table 4: The results of comparison between MiniMoE and baselines upon distilling $\text{BERT}_{\text{base}}$. The best results are **boldfaced**.

Method	GFLOPs		SST-2 Acc	MRPC F1	STS-B SpCorr	QQP F1	MNLI-m/mm Acc	QNLI Acc	RTE Acc	GLUE Score	CoNLL F1
$\text{BERT}_{\text{base}}$	10.9		93.8	91.5	87.1	88.4	84.9/84.9	91.9	71.5	86.7	94.8
$\text{KD}_{15\%}$	1.64	6~8×	89.9	88.6	85.1	86.2	79.8/80.2	85.6	63.9	82.4	92.8
$\text{PKD}_{15\%}$	1.64		90.0	88.2	85.5	86.4	80.4/79.6	85.9	63.9	82.5	92.9
$\text{MoEBERT}_{17\%}$[1]	1.86		89.6	88.4	85.1	86.8	80.4/80.5	86.6	65.0	82.8	92.7
$\text{DynaBERT}_{15\%}$[2]	1.64		89.1	85.1	84.7	84.3	78.3/79.0	86.6	61.4	81.1	-
$\text{MiniDisc}_{15\%}$[3]	1.64		89.8	88.2	85.8	86.6	80.3/79.9	87.3	68.2	83.3	93.0
$\text{MiniLM}_{6L;384H}$	1.36		91.1	90.1	88.1	86.7	81.5/**81.8**	89.2	67.9	84.5	93.2
w/ TA	1.36		91.3	**90.3**	88.2	**86.8**	81.4/81.6	**89.7**	66.8	84.5	93.2
$\text{MiniMoE}_{6L;384H}$	1.36		**91.3**	90.2	**88.6**	86.5	**81.6**/81.5	89.5	**68.6**	**84.7**	**93.3**
$\text{KD}_{10\%}$	1.08	10~12×	88.2	87.6	84.0	84.4	77.6/77.4	84.3	67.2	81.3	91.2
$\text{MiniDisc}_{10\%}$	1.08		89.1	88.4	85.4	84.9	78.2/78.6	86.3	68.2	82.4	91.9
$\text{MiniLM}_{4L;384H}$	0.91		90.0	**88.6**	87.2	86.1	80.0/80.3	87.9	67.2	83.4	91.5
w/ TA	0.91		90.0	88.5	87.3	**86.3**	**80.1/80.7**	88.0	66.4	83.4	91.8
$\text{MiniMoE}_{4L;384H}$	0.91		**90.8**	88.1	**88.2**	85.9	79.8/80.4	**88.6**	**69.3**	**83.9**	**92.3**
$\text{KD}_{5\%}$	0.54	16~20×	85.6	84.0	83.8	82.5	72.6/73.2	81.6	63.2	78.3	83.1
$\text{MiniDisc}_{5\%}$	0.54		86.9	87.6	84.8	83.5	72.7/74.5	84.0	66.8	80.1	85.6
$\text{TinyBERT}_{4L;312H}$[4]	0.60		88.5	87.9	86.6	85.6	**78.9/79.2**	**87.3**	**67.2**	**82.7**	-
$\text{MiniLM}_{3L;384H}$	0.68		89.1	**89.1**	86.6	85.4	77.8/78.4	87.2	66.1	82.5	90.1
w/ TA	0.68		**89.8**	87.8	86.0	85.5	77.6/78.5	86.8	66.1	82.3	90.4
$\text{MiniMoE}_{3L;384H}$	0.68		89.3	87.4	**87.8**	**85.6**	78.2/78.7	87.2	67.0	82.6	**90.7**
$\text{KD}_{3\%}$	0.32	34~47×	85.2	83.6	81.9	82.1	71.9/72.7	81.9	57.4	77.1	74.3
$\text{MiniDisc}_{3\%}$	0.32		85.9	85.7	83.6	83.1	72.9/73.6	81.9	58.1	78.1	80.5
$\text{MiniLM}_{4L;192H}$	0.23		86.9	**86.4**	85.4	84.3	77.5/77.5	85.9	65.3	81.2	90.0
w/ TA	0.23		87.2	85.6	86.2	84.6	77.3/**78.0**	86.6	64.6	81.3	89.9
$\text{MiniMoE}_{4L;192H}$	0.23		**88.1**	86.1	**86.2**	**84.8**	**77.7**/77.8	**86.6**	**68.6**	**82.0**	**91.3**

[1] Each FFN is split to 8 experts and each MHA to 4 to reach the sparsity.
[2] The results are produced from the released code.
[3] The results are mainly taken from the original papers.
[4] The results are produced without additional task-specific distillation.

performs $\text{TinyBERT}_{4L;312H}$ is conjectured due to structure discrepancy. Another observation is that the larger times of compression, the larger the performance improvements are. For example, $\text{MiniMoE}_{4L;384H}$ yields an absolute 0.5 performance improvement over $\text{MiniLM}_{4L;384H}$ in contrast to that $\text{MiniMoE}_{6L;384H}$ only has an absolute 0.2 performance improvement over $\text{MiniLM}_{6L;384H}$ on GLUE. Two more notes are that, MoEBERT nearly reaches the compression upper bound, and TinyBERT is reproduced without additional task-specific distillation for a fair comparison while the results with additional task-specific distillation are supplied in Appendix C.

4.5 Analyses

Practical Inference Compute Since GFLOPs can only measure the theoretical inference compute, we further provide throughput (i.e., tokens per micro second) as a practical inference compute measure. As in Table 5, 20× compression can realize a significant inference compute gain in comparing $\text{KD}_{5\%}$ to $\text{BERT}_{\text{base}}$. The practical speedup is approximately 6.7×. Moreover, $\text{MiniMoE}_{3L;384H}$ can retain most inference compute gain even if the routing algorithm can slightly reduce the gain when compared to $\text{MiniLM}_{3L;384H}$. Although MiniMoE is seemingly memory-inefficient regarding the increased parameter amount, we argue the potential of a memory-efficient MiniMoE with parameter decomposition in Appendix G.

Table 5: Practical inference compute with reference to $\text{BERT}_{\text{base}}$.

Method	GFLOPs	Throughput	Params
$\text{BERT}_{\text{base}}$	10.9	80.8 tokens/ms	109.5 M
$\text{KD}_{5\%}$	0.54	544.7 tokens/ms	28.7 M
$\text{MiniLM}_{3L;384H}$	0.68	485.3 tokens/ms	17.2 M
$\text{MiniMoE}_{3L;384H}$	0.68	433.1 tokens/ms	28.3 M

Student Scale Following the behavior of Figure 2, we would like to showcase whether MiniMoE can lift the curse across difference student scales. From Figure 4, the curse is lifted to a large extent by MiniMoE in comparison with MiniLM and MiniLM w/ TA. However, MiniMoE meets a bottleneck that distilling $\text{BERT}_{\text{large}}$ makes no difference from distilling $\text{BERT}_{\text{base}}$ when the FLOPs is at an extreme value 0.04G ($\sim$273$\times$ compression from $\text{BERT}_{\text{base}}$, $\sim$968$\times$ compression from $\text{BERT}_{\text{large}}$). We explore the extreme case by plugging a TA to MiniMoE as supplied in Appendix D.

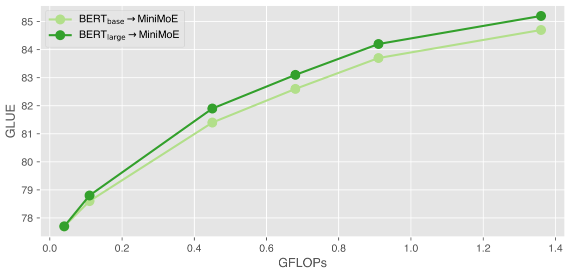

Figure 4: The performance of MiniMoE across different student scales upon distilling $\text{BERT}_{\text{base}}$.

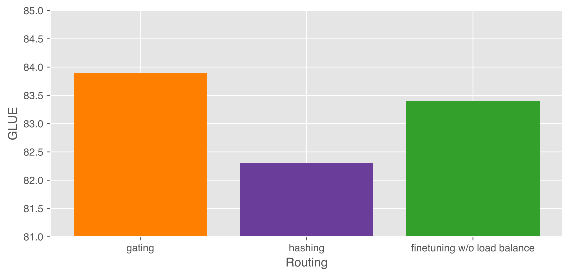

Figure 5: The performance of different routing choices with $\text{MiniMoE}_{\text{4L;384H}}$ upon distilling $\text{BERT}_{\text{base}}$.

Routing Algorithm Routing algorithm is also a crucial part benefiting from a nice design choice. We compare our used gating with another fancy choice hashing. We at the same time show the effect of using load balance at finetuning stage as well. From the results in Figure 5, we see that gating outperforms hashing, and load balancing at both distillation and finetuning stages is superior to that at only distillation stage.

Expert Number Regarding the expert number m is a parameter of great importance for MiniMoE, we here study its impact on the performance. The results in Figure 6 reveal a first ascending then descending phenomenon while adding experts at a time. The phenomenon suggests there is a tradeoff when increasing the number of experts, and we conjecture the tradeoff accords with the famous bias-variance tradeoff (Hastie et al., 2001, Chapter 7). That is, adding experts grows the parameter scale, thus decreasing bias yet increasing variance. Another interesting notice is that smaller students favor fewer experts. Based on the tradeoff conjecture, we hypothesize that smaller students are more sensitive to variance increment, as the biases of smaller students can arrive at a minimum more quickly than those of larger ones.

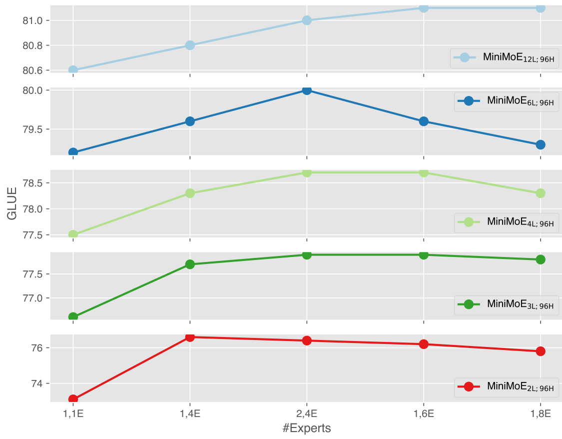

Figure 6: The impact of expert number on the performance upon distilling $\text{BERT}_{\text{base}}$, where x,yE denotes x experts in each MHA and y experts in each FFN. For example, 1,1E is the original dense model, and 1,4E is the MoE model used in Table 4.

5 Conclusions

In this work, we uncover a curse of capacity gap in LM distillation, which is well discussed in previous studies on vision model distillation but not recognized in distilling LMs. While there are some studies investigating to fill the gap, we find they can hardly tackle the curse. Interestingly, existing solutions in large vision language model distillation which are stated to be able to lift the curse fail to achieve so for LMs. So we aim at lifting the curse by proposing a well-motivated MiniMoE. The MiniMoE can essentially enlarge the capacity of the student but leave the inference compute nearly untouched. Our experimental results indicate that MiniMoE can not only lift the curse but also realize new state of the arts.

Limitations

The central limitation of MiniMoE is the increased memory footprint, which we could potentially address in the near future according to Appendix G.

Acknowledgements

We thank the anonymous reviewers and chairs for their constructive suggestions. This research was supported in part by Natural Science Foundation of Beijing (grant number: 4222036) and Huawei Technologies (grant number: TC20201228005). Jingang Wang is funded by Beijing Nova Program (grant number: 20220484098).

References

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael R. 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4334–4348.

Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. 2015. Conditional computation in neural networks for faster models. *arXiv*, 1511.06297.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2011. The seventh PASCAL recognizing textual entailment challenge. In *Proceedings of the Fourth Text Analysis Conference, TAC 2011, Gaithersburg, Maryland, USA, November 14-15, 2011*.

Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, pages 535–541.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 1–14.

Jang Hyun Cho and Bharath Hariharan. 2019. On the efficacy of knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 4793–4801.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways. *arXiv*, abs/2204.02311.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv*, abs/2210.11416.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing, IWP@IJCNLP 2005, Jeju Island, Korea, October 2005, 2005*.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P. Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen S. Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. Glam: Efficient scaling of language models with mixture-of-experts. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569.

William Fedus, Barret Zoph, and Noam Shazeer. 2021. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *arXiv*, 2101.03961.

Saurabh Goyal, Anamitra Roy Choudhury, Saurabh Raje, Venkatesan T. Chakaravarthy, Yogish Sabharwal, and Ashish Verma. 2020. Power-bert: Accelerating BERT inference via progressive word-vector elimination. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 3690–3699.

Song Han, Jeff Pool, John Tran, and William J. Dally. 2015. Learning both weights and connections for efficient neural networks. *arXiv*, 1506.02626.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. 2021. Dynamic neural networks: A survey. *arXiv*, 2102.04906.

Trevor Hastie, Jerome H. Friedman, and Robert Tibshirani. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics.

Jiaao He, Jiezhong Qiu, Aohan Zeng, Zhilin Yang, Jidong Zhai, and Jie Tang. 2021. Fastmoe: A fast mixture-of-expert training system. *arXiv*, 2103.13262.

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *arXiv*, 1503.02531.

Lu Hou, Zhiqi Huang, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2020. Dynabert: Dynamic BERT with adaptive width and depth. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

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, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 4163–4174.

Gyuwan Kim and Kyunghyun Cho. 2021. Length-adaptive transformer: Train once with length drop, use anytime with search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6501–6511.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*.

Jianquan Li, Xiaokang Liu, Honghong Zhao, Ruifeng Xu, Min Yang, and Yaohong Jin. 2020. BERT-EMD: many-to-many layer mapping for BERT compression with earth mover's distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3009–3018.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv*, 1907.11692.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. 2016. Unifying distillation and privileged information. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. 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 5191–5198.

Dae Young Park, Moon-Hyun Cha, Changwook Jeong, Daesin Kim, and Bohyung Han. 2021a. Learning student-friendly teacher networks for knowledge distillation. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 13292–13303.

Geondo Park, Gyeongman Kim, and Eunho Yang. 2021b. Distilling linguistic context for language model compression. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 364–378.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:140:1–140:67.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation AI scale. In *International Conference on Machine*

Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA, volume 162 of *Proceedings of Machine Learning Research*, pages 18332–18346.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392.

Stephen Roller, Sainbayar Sukhbaatar, Arthur Szlam, and Jason Weston. 2021. Hash layers for large sparse models. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 17555–17566.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv*, 1910.01108.

Noam Shazeer, Youlong Cheng, Niki Parmar, Dustin Tran, Ashish Vaswani, Penporn Koanantakool, Peter Hawkins, HyoukJoong Lee, Mingsheng Hong, Cliff Young, Ryan Sepassi, and Blake A. Hechtman. 2018. Mesh-tensorflow: Deep learning for supercomputers. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10435–10444.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. 2019. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv*, abs/1909.08053.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1631–1642.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 4322–4331.

Siqi Sun, Zhe Gan, Yuwei Fang, Yu Cheng, Shuohang Wang, and Jingjing Liu. 2020a. Contrastive distillation on intermediate representations for language model compression. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 498–508.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020b. Mobilebert: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2158–2170.

Wonyong Sung, Sungho Shin, and Kyuyeon Hwang. 2015. Resiliency of deep neural networks under quantization. *arXiv*, 1511.06488.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *arXiv*, 1908.08962.

Vladimir Vapnik. 1998. *Statistical learning theory*. Wiley.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Shuohuan Wang, Yu Sun, Yang Xiang, Zhihua Wu, Siyu Ding, Weibao Gong, Shikun Feng, Junyuan Shang, Yanbin Zhao, Chao Pang, Jiaxiang Liu, Xuyi Chen, Yuxiang Lu, Weixin Liu, Xi Wang, Yangfan Bai, Qiuliang Chen, Li Zhao, Shiyong Li, Peng Sun, Dianhai Yu, Yanjun Ma, Hao Tian, Hua Wu, Tian Wu, Wei Zeng, Ge Li, Wen Gao, and Haifeng Wang. 2021a. ERNIE 3.0 titan: Exploring larger-scale knowledge enhanced pre-training for language understanding and generation. *arXiv*, 2112.12731.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021b. Minilmv2: Multi-head self-attention relation distillation for compressing pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2140–2151.

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 *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions on Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel R. Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1112–1122.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 1513–1528.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. Deebert: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 2246–2251.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, Yina Patterson, Zuoyu Tian, Yiwen Zhang, He Zhou, Shaoweihua Liu, Zhe Zhao, Qipeng Zhao, Cong Yue, Xinrui Zhang, Zhengliang Yang, Kyle Richardson, and Zhenzhong Lan. 2020. CLUE: A chinese language understanding evaluation benchmark. In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 4762–4772. International Committee on Computational Linguistics.

Fuzhao Xue, Xiaoxin He, Xiaozhe Ren, Yuxuan Lou, and Yang You. 2022. One student knows all experts know: From sparse to dense. *arXiv*, 2201.10890.

Yi Yang, Chen Zhang, and Dawei Song. 2022. Sparse teachers can be dense with knowledge. *arXiv*, abs/2210.03923.

Sha Yuan, Hanyu Zhao, Zhengxiao Du, Ming Ding, Xiao Liu, Yukuo Cen, Xu Zou, Zhilin Yang, and Jie Tang. 2021. Wudaocorpora: A super large-scale chinese corpora for pre-training language models. *AI Open*, 2:65–68.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8BERT: quantized 8bit BERT. In *Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition, EMC2@NeurIPS 2019, Vancouver, Canada, December 13, 2019*, pages 36–39.

Chen Zhang, Yang Yang, Qifan Wang, Jiahao Liu, Jingang Wang, Yunsen Xian, Wei Wu, and Dawei Song. 2022a. Minidisc: Minimal distillation schedule for language model compression. *arXiv*, 2205.14570.

Zhengyan Zhang, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2022b. Moefication: Transformer feed-forward layers are mixtures of experts. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 877–890.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. 2022. Decoupled knowledge distillation. *arXiv*, 2203.08679.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian J. McAuley, Ke Xu, and Furu Wei. 2020. BERT loses patience: Fast and robust inference with early exit. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wangchunshu Zhou, Canwen Xu, and Julian J. McAuley. 2022. BERT learns to teach: Knowledge distillation with meta learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7037–7049.

Yichen Zhu and Yi Wang. 2021. Student customized knowledge distillation: Bridging the gap between student and teacher. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 5037–5046.

Simiao Zuo, Qingru Zhang, Chen Liang, Pengcheng He, Tuo Zhao, and Weizhu Chen. 2022. Moebert: from BERT to mixture-of-experts via importance-guided adaptation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 1610–1623.

A Data Summary

The detailed statistics, maximum sequence lengths, and metrics for datasets we use are shown in Table 6, where the Wikipedia corpus used for distillation is also attached.

B More Hands-on Details

General Guidelines The details of hyperparameters for distillation and finetuning are shown in Table 7. We will be releasing our code and scripts in the final version for exact reproducibility. For all cases, students are always randomly initialized following MiniLM.

Implementation of MiniMoE We strictly follow the design of SwitchTransformer (Fedus et al., 2021) and extend it to the design of our MINIMOE. We also follow their associated appendices to implement an MoE for multihead attention. In detail, based on the original design, we treat an FFN/MHA as an minimal expert, adopt top-*one* gating with load balancing, and employ a capacity factor of 1.25 for a good tradeoff (where overflowed tokens are dropped). For the parameter effect of adding an expert, we take expanding MiniLM$_{\text{4L;192H}}$ (11.3M) to MiniMoE$_{\text{4L;192H}}$-1,2E (14.9M) as an example. The number of parameters for embeddings is not changed (6.0M$\rightarrow$6.0M), but adding an expert (1,1E$\rightarrow$1,2E) results in an increased number of parameters for transformers (5.4M$\rightarrow$9.0M).

Further, our design for HashLayer (Roller et al., 2021) also strictly follows the original random hash design, i.e., per-token hash is used. We strictly follow the best configuration of DeKD as reported in their paper (Zhao et al., 2022), where α is 1.0 and β is 8.0.

C Results w/ Task-specific Distillation

The results with task-specific distillation are produced from released checkpoints. The results in Table 8 demonstrate that TinyBERT is largely supported with data augmentation in the task-specific distillation stage for great performance. Another intriguing observation is that data augmentation only works for distillation but not for finetuning potentially due to the noise-resilience of distillation, so we preferably replace the finetuning stage with a task-specific distillation stage in experimenting with MiniLM.

D MINIMOE at Extreme

The results in Table D witness that, MINIMOE sometimes struggles with extreme cases but can be enhanced with the help of TA.

E Related Work

Knowledge Distillation Distillation (Hinton et al., 2015) is a de facto way to compression (Bucila et al., 2006) LMs by transferring the knowledge of LMs to small language models. During the distillation, a small language model serves as a student and treats a LM as a teacher to learn from. There are three lines of work in LM distillation: firstly, task-specific distillation (Sun et al., 2019; Li et al., 2020; Sun et al., 2020a; Park et al., 2021b; Hou et al., 2020; Xia et al., 2022) that conducts distillation on a specific task at finetuning stage; secondly, task-agnostic distillation (Turc et al., 2019; Sanh et al., 2019; Sun et al., 2020b; Wang et al., 2021b) that conducts distillation at pretraining stage; and thirdly, two-stage distillation (Jiao et al., 2020) that combines the power of both task-agnostic and -specific distillation. Though these methods realize promising performance when distilling LMs like BERT$_{\text{base}}$, they can come short of scalability to LMs like BERT$_{\text{large}}$ especially when the student is of a small scale. In fact, driven by recent observations (Wang et al., 2020; Zhang et al., 2022a; Mirzadeh et al., 2020; Cho and Hariharan, 2019), distillation with a small student can be faced with two deficiencies due to the large capacity gap. A few studies including teacher assistant-based (Mirzadeh et al., 2020; Zhang et al., 2022a) and student-friendly (Park et al., 2021a; Zhou et al., 2022) distillation can alleviate the first but fail to resolve the second. It is noteworthy that some work states they can tackle both deficiencies for vision models (Zhu and Wang, 2021; Zhao et al., 2022), but preliminary studies have found that they are either expensive or not capable of LMs. In our work, we follow the line of task-agnostic distillation of LMs and aims at lifting both efficiencies for the first time.

Mixture of Experts Based on the idea of conditional computation (Bengio et al., 2015), MoE layer is proposed to scale-up LMs in a sparsely activated fashion (Shazeer et al., 2017). There are diverse designs to achieve the sparse routing, such as gating (Shazeer et al., 2018) and hashing (Roller et al., 2021), with necessary balance constraints (Lepikhin et al., 2021). MoE layers are then joined to LMs in the past one or two years (Fedus et al., 2021; Du et al., 2022). Owing to the sparse activation property, the scales of LMs are significantly

Table 6: The statistics, maximum sequence lengths, and metrics.

Dataset	#Train exam.	#Dev exam.	Max. length	Metric
SST-2	67K	0.9K	64	Accuracy
MRPC	3.7K	0.4K	128	F1
STS-B	7K	1.5K	128	Spearman Correlation
QQP	364K	40K	128	F1
MNLI-m/mm	393K	20K	128	Accuracy
QNLI	105K	5.5K	128	Accuracy
RTE	2.5K	0.3K	128	Accuracy
CoNLL	14k	3.3k	128	F1
Wikipedia	35M	-	128	-

Table 7: The hyperparameters for both distillation and finetuning. The search grids for GLUE and CoNLL are indicated differently.

Hyperparameter	Distillation	Finetuning
Batch size	$8\times128=1024$	{16,32}
Optimizer	AdamW	AdamW
Learning rate	3e-4	{1e-5,2e-5,3e-5}/{1e-4,2e-4,3e-4}
Training epochs	5	10
Earlystop epochs	-	5
Warmup proportion	0.01	0.1
Weight decay	0.01	0.01

increased with only minor losses in compute efficiency on modern GPU devices so that the underneath scaling laws can be uncovered in a comparably cheap manner (He et al., 2021; Rajbhandari et al., 2022). In our work, we are impelled by the merits of MoE, and propose a MINIMOE so that the capacity of the student can be enlarged without much inference overhead increment. MINIMOE can be similar to a certain stream of methods (Zhang et al., 2022b; Zuo et al., 2022) that pursue accelerating LMs via precisely moefying them. Nonetheless, the moefication process is exerted to LMs with limited inference compute improvements compared to those advanced by MINIMOE. Note that there are emergent work exploring compressing MoE LMs (Xue et al., 2022) to dense students, which is walking down the same street in the opposite side since we instead focus on compressing dense LMs to MoE students.

F Results on BERT$_{\text{xlarge}}$

LM distillation, under either the task-agnostic setting as in our paper or the task-specific setting, has seldom been investigated to distil LMs larger than BERT$_{\text{large}}$. Even worse, there is only little work has been investigated to distil BERT$_{\text{large}}$ under the task-agnostic setting.

In the main results, we just follow the paces of the task-agnostic setting, not only due to the huge scales of larger LMs like T5 and GPT3 but also due to that task-agnostic LM distillation requires the access to the original pretraining data of usually vast volume. What's more, larger LMs like T5 can be incomparable to BERT owing to the architectural difference, and existing task-agnostic methods including ours may easily fail.

Regarding all the considerations mentioned above, however, we try to check the existence of the curse of capacity gap and examine MINIMOE under a comparably larger-scale setting, i.e., Chinese BERT$_{\text{base}}$ v.s. BERT$_{\text{xlarge}}$ on some datasets from CLUE (Xu et al., 2020) (which can be viewed as the Chinese GLUE). These datasets include a topic classification dataset TNews, a similar question matching dataset AFQMC, and a natural language inference dataset OCNLI. The preliminary results are shown in Table 10. As far as we know, while English BERT$_{\text{xlarge}}$ with more than one billion parameters trained by Nvidia Megatron (Shoeybi et al., 2019) is not publicly available, Chinese BERT$_{\text{xlarge}}$ can be easily downloaded through hug-

Table 8: The results with and without task-specific distillation upon distilling BERT$_{\text{base}}$.

Method	GFLOPs	SST-2 Acc	MRPC F1	STS-B SpCorr	QQP F1	MNLI-m/mm Acc	QNLI Acc	RTE Acc	GLUE Score
TinyBERT$_{\text{4L;312H}}$	0.60	88.5	87.9	86.6	85.6	78.9/79.2	87.3	67.2	82.7
w/ tsd.+aug.	0.60	91.6	90.2	86.3	87.1	81.2/82.8	87.6	64.3	83.9
MiniDisc$_{5\%}$	0.54	86.9	87.6	84.8	83.5	72.7/74.5	84.0	66.8	80.1
w/ aug.	0.54	91.2	90.0	87.5	85.4	79.0/79.8	84.5	67.5	83.1
MiniLM$_{\text{3L;384H}}$	0.68	89.1	89.1	86.6	85.4	77.8/78.4	87.2	66.1	82.5
w/ aug.	0.68	88.7	85.9	83.1	82.8	76.2/76.0	86.6	62.5	80.2
w/ tsd.+aug.*	0.68	91.2	91.1	88.2	86.6	79.9/80.4	87.8	66.1	83.9

* tsd. indicates task-specific distillation and aug. indicates distillation with data augmentation.

Table 9: The results of MiniMoE at extreme upon distilling BERT$_{\text{base}}$ and BERT$_{\text{large}}$ respectively.

Method	GFLOPs		SST-2 Acc	MRPC F1	STS-B SpCorr	QQP F1	MNLI-m/mm Acc	QNLI Acc	RTE Acc	GLUE Score
BERT$_{\text{base}}$	10.9		93.8	91.5	87.1	88.4	84.9/84.9	91.9	71.5	86.7
MiniLM$_{\text{4L;96H}}$	0.06	~182×	83.4	84.6	81.9	80.7	71.2/72.5	82.0	63.7	77.5
w/ TA	0.06		84.5	83.9	82.2	80.5	70.8/72.4	81.6	63.7	77.5
MiniMoE$_{\text{4L;96H}}$	0.06		84.8	84.0	83.1	81.2	72.2/73.5	82.2	65.7	78.3
w/ TA	0.06		84.2	85.3	83.7	82.2	72.6/73.7	83.6	65.3	78.8
MiniLM$_{\text{3L;96H}}$	0.04	~273×	83.7	83.8	81.2	80.6	70.3/71.5	80.5	61.4	76.6
w/ TA	0.04		82.6	83.3	81.2	80.3	70.3/71.9	80.7	61.4	76.5
MiniMoE$_{\text{3L;96H}}$	0.04		84.8	84.5	82.8	80.8	70.3/71.9	81.9	65.0	77.7
w/ TA	0.04		83.5	85.1	83.1	81.4	71.4/73.0	83.3	61.7	77.8
BERT$_{\text{large}}$	38.7		94.2	92.5	90.1	89.0	86.6/86.3	92.5	75.5	88.3
MiniLM$_{\text{4L;96H}}$	0.06	~645×	83.3	83.9	82.5	81.0	71.4/72.4	81.8	63.2	77.4
w/ TA	0.06		84.1	85.8	82.4	81.3	71.9/73.4	82.3	64.3	78.2
MiniMoE$_{\text{4L;96H}}$	0.06		84.9	85.4	82.9	81.6	74.0/74.8	83.6	64.6	79.0
w/ TA	0.06		84.2	85.3	83.2	81.2	72.5/74.0	83.4	66.1	78.7
MiniLM$_{\text{3L;96H}}$	0.04	~968×	83.1	84.1	81.8	79.7	69.7/70.8	79.2	63.2	76.5
w/ TA	0.04		83.0	83.2	81.2	80.3	69.3/70.7	81.8	60.7	76.3
MiniMoE$_{\text{3L;96H}}$	0.04		83.0	84.5	82.7	81.1	71.7/72.8	82.1	63.9	77.7
w/ TA	0.04		83.8	84.4	83.0	81.2	71.8/72.8	82.4	63.9	77.9

gingface.[2] It is noteworthy that Chinese BERT$_{\text{base}}$ is trained on Chinese Wikipedia (∼15G) while Chinese BERT$_{\text{xlarge}}$ is trained on Wudao Corpus (∼300G) (Yuan et al., 2021). We use Wikipedia data as the default choice for distillation, but Wudao data seems to be a more suitable (though not that fair) one for distilling Chinese BERT$_{\text{xlarge}}$ as we have found that Wikipedia could not make the distillation converge properly. Painfully, it consumes around one week to achieve one epoch of distilling Chinese BERT$_{\text{xlarge}}$ using Wudao in contrast to five epochs of distilling Chinese BERT$_{\text{base}}$ on Wikipedia in one day. The results show that Chinese BERT$_{\text{xlarge}}$ is cursed to realize better students than Chinese BERT$_{\text{base}}$ does, and MiniMoE has the potential to lift the curse under the larger-scale setting.

[2] https://huggingface.co/IDEA-CCNL/Erlangshen-MegatronBert-1.3B.

G Potential of Memory-efficient MiniMoE

One may argue that MiniMoE introduces much more memory consumption than MiniLM does, largely limiting the application scenarios for memory-sensitive devices (e.g., mobile devices).

However, there is no free lunch to enlarge the capacity of the student. We should claim that, in order to increase the capacity, memory/space consumption is a cheaper choice (e.g., more experts) than latency/time consumption (e.g., more operations), and this is potentially the reason why

Table 10: The results of comparison between distilling Chinese $\text{BERT}_{\text{base}}$ and $\text{BERT}_{\text{xlarge}}$.

Method	**Teacher**	**TNews Acc**	**AFQMC Acc**	**OCNLI Acc**	**CLUE Score**
Teacher	$\text{BERT}_{\text{base}}$	57.0	74.8	75.4	69.1
	$\text{BERT}_{\text{xlarge}}\Uparrow$	60.0	76.1	79.2	71.7
$\text{MiniLM}_{\text{6L;384H}}$	$\text{BERT}_{\text{base}}$	55.5	72.0	71.0	66.2
	$\text{BERT}_{\text{xlarge}}\Downarrow$	54.9	70.7	69.9	65.2
$\text{MiniMoE}_{\text{6L;384H}}$	$\text{BERT}_{\text{base}}$	55.9	72.9	70.8	66.5
	$\text{BERT}_{\text{xlarge}}\Uparrow$[1]	56.7	72.4	71.0	66.7

[1] $\Uparrow$ is used to indicate the deficiency is tackled on CLUE, otherwise $\Downarrow$ is used.

large LMs like PaLM (Chowdhery et al., 2022) and FLAN (Chung et al., 2022) could become so popular. We should also highlight that scenarios that require rather limited memory consumption (e.g., mobile scenarios) is currently not (though can be in the near future) the main concern of LMs. In contrast, LMs are usually served in GPU scenarios, where memory/space is easy to access.

Luckily, we find a potential path to address the memory efficiency concern based on the idea of parameter decomposition (e.g., SVD). While embedding parameter decomposition is a general way to reduce the number of parameters for embeddings and could not make MiniMoE as memory-efficient as MiniLM. We uncover that, without much performance sacrifice, transformer parameter decomposition in MiniMoE can be easier in comparison with that in MiniLM owing to the sparse activation property of MoE. That is, transformer parameters in MiniMoE have lower ranks than those in MiniLM, and this can be shown by analyzing the magnitudes of the normalized singular values using SVD. The preliminary results of the output matrices of the last FFN layers separately from $\text{MiniLM}_{\text{3L;384H}}$ and $\text{MiniMoE}_{\text{3L;384H}}$ are shown in Table 11.

With this finding, MiniMoE can compress more parameters than MiniLM does using parameter decomposition and finally yield a similar memory efficiency to that of MiniLM.

We also explore another complementary solution that views MiniMoE as teacher assistant and further distils from MiniMoE to its dense counterpart. The results are shown in Table 12, implying that this only results in an acceptable performance degradation on large datasets like MNLI and SST-2 but undesired performance degradation on small datasets like RTE.

H Failure of Vision Method

We examine in a preliminary study the effectiveness of one of the vision model distillation methods (DeKD, Zhao et al., 2022) which can lift the curse of capacity gap. From the results in Table 13, we unfortunately discover that DeKD can only give comparable performance in distilling $\text{BERT}_{\text{base}}$, which even lags behind KD w/ TA. It hints that vision model distillation methods are not that capable of LMs.

Table 11: The SVD analysis to show the potential of memory-efficient MINIMOE.

Method	**%Value>0.2**	**%Value>0.1**	**%Value>0.05**	**Trm Params (Value>0.1)**
MiniLM$_{\text{3L;384H}}$ dense	315/384=82%	356/384=93%	373/384=97%	5.3M→5.1M
MiniMoE$_{\text{3L;384H}}$ expert #1	6/384=2%	82/384=21%	275/384=72%	-
MiniMoE$_{\text{3L;384H}}$ expert #2	34/384=9%	220/384=57%	361/384=94%	-
MiniMoE$_{\text{3L;384H}}$ expert #3	15/384=4%	175/384=46%	338/384=88%	-
MiniMoE$_{\text{3L;384H}}$ expert #4	24/384=6%	200/384=52%	357/384=93%	-
MiniMoE$_{\text{3L;384H}}$ all experts	79/384/4=5%	677/384/4=44%	1331/384/4=87%	16.4M→8.2M

Table 12: The results of further distilling from MINIMOE to its dense counterpart.

Method	**MNLI-m/mm**	**SST-2**	**RTE**
MiniLM$_{\text{3L;384H}}$	77.8/78.4	89.1	66.1
MINIMOE$_{\text{3L;384H}}$	78.2/78.7	89.3	67.0
MINIMOE$_{\text{3L;384H}}$⇒MiniLM$_{\text{3L;384H}}$	78.1/78.4	89.5	64.3

Table 13: The results of applying vision distillation methods upon BERT$_{\text{base}}$.

Method	**GLUE**	**Method**	**GLUE**
KD$_{\text{2L}}$	72.9	KD$_{\text{4L}}$	81.8
w/ TA	73.4	w/ TA	82.1
DeKD$_{\text{2L}}$	72.7	DeKD$_{\text{4L}}$	81.6

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
the section following the conclusions.

☒ A2. Did you discuss any potential risks of your work?
not any known risks.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
the introduction.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☒ Did you use or create scientific artifacts?

Left blank.

☐ B1. Did you cite the creators of artifacts you used?
No response.

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

☐ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

the experiments.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
the results in the experiments.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
the hands-on details in the experiments.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
the results in the experiments.

☒ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
not related packages used.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.