How to Trade Off the Quantity and Capacity of Teacher Ensemble: Learning Categorical Distribution to Stochastically Employ A Teacher for Distillation

Zixiang Ding[*1], **Guoqing Jiang**[1], **Shuai Zhang**[1], **Lin Guo**[1], **Wei Lin**[2]
[1]Meituan
[2]Independent Researcher
{dingzixiang, jiangguoqing03, zhangshuai51}@meituan.com, lguo7@jhu.edu, lwsaviola@163.com

Abstract

We observe two phenomenons with respect to quantity and capacity: 1) more teacher is not always better for multi-teacher knowledge distillation, and 2) stronger teacher is not always better for single-teacher knowledge distillation. To trade off the quantity and capacity of teacher ensemble, in this paper, we propose a new distillation paradigm named Dynamic Knowledge Distillation (DynaKD) that learn an adaptive categorical distribution to stochastically employ a teacher from a teacher ensemble in each step, to transfer knowledge from teacher ensemble into student. DynaKD has three advantages: 1) it can preserve diversity of each teacher via one-to-one distillation manner instead of several-for-one, 2) it can make the best of powerful teacher via those multi-level assistant teachers in ensemble, and 3) it can also dynamically determine the importance of each teacher for various tasks. To verify the effectiveness of the proposed approach, we conduct extensive experiments for BERT compression on GLUE benchmark. Experimental results show that the proposed approach achieves state-of-the-art score compared to previous compression approaches on five out of seven downstream tasks, including pushing MRPC F1 and accuracy to 92.2 (1.4 point absolute improvement), RTE accuracy to 76.2 (2.8 point absolute improvement). Moreover, we conduct also extensive experiments for image classification on CIFAR-100. Similarly, DynaKD achieves also state-of-the-art performance.

Introduction

BERT (Devlin et al. 2019) has brought about a sea change for natural language processing. Following BERT, numerous subsequent works focus on various perspectives to further improve its performance, e.g., hyper-parameter (Liu et al. 2019b), learnable embedding paradigm (Raffel et al. 2020), architecture (Gao et al. 2022), etc. However, there are massive redundancies in the above BERT-style models w.r.t. attention heads (Dong, Cordonnier, and Loukas 2021), weights (Gordon, Duh, and Andrews 2020), and layers (Fan, Grave, and Joulin 2020). Consequently, many compact BERT-style language models are proposed via pruning (Fan, Grave, and Joulin 2020), quantization (Shen et al. 2020), parameter sharing (Lan et al. 2020) and Knowledge Distillation (KD) (Pan et al. 2021). In this paper, we focus on the KD-based compression approaches.

From the point of view of learning procedure, KD is used in both pre-training (Turc et al. 2019; Sanh et al. 2019; Sun et al. 2020; Jiao et al. 2020) and fine-tuning phases (Jiao et al. 2020; Wu, Wu, and Huang 2021; Ding et al. 2023). From the point of view of distillation objective, KD is employed for the outputs of hidden layer (Sun et al. 2020), final layer (Wu, Wu, and Huang 2021), embedding (Sanh et al. 2019) and self-attention (Wang et al. 2020). Wu, Wu, and Huang (2021) employ multiple teachers to achieve better performance than single-teacher KD based approaches on several downstream tasks of GLUE benchmark. Nevertheless, the ensemble of multiple teachers is not always more effective than the single teacher for student distillation (see Table 3), where same observations are also existed in SKDBERT (Ding et al. 2023). There are two possible reasons as mentioned in SKDBERT: 1) *diversity losing* (Tran et al. 2020) and 2) *underutilization of powerful teacher* (Mirzadeh et al. 2020). To solve this issue, Ding et al. (2023) propose a distillation strategy named SKD where a teacher is stochastically sampled from a teacher ensemble according to a handcrafted distribution. However, it is very time-consuming to design appropriate sampling distribution.

Inspired by SKDBERT, as shown in Figure 1, we propose DynaKD which stochastically employs a teacher from a multi-level teacher ensemble following a learned adaptive categorical distribution with regard to different downstream tasks in each step. The proposed approach can not only solve the above mentioned issues, but also automate the designing process of sampling distribution (dubbed as categorical distribution in this paper). Furthermore, we also propose a differentiable algorithm to dynamically determine the adaptive categorical distribution which plays an important role in DynaKD. We implement extensive experiments on GLUE benchmark to verify the effectiveness of DynaKD. Moreover, to show the generalization capacity, we have also distilled deep convolutional neural network by DynaKD for computer vision on CIFAR-100 (Krizhevsky, Hinton et al. 2009). Our contributions are summarized as follows:

- We propose DynaKD which preserves the diversity of each teacher in local view (i.e., each step) and make the best use of strong teachers in global view (i.e., entire process) under an adaptive categorical distribution.

[*]Corresponding author

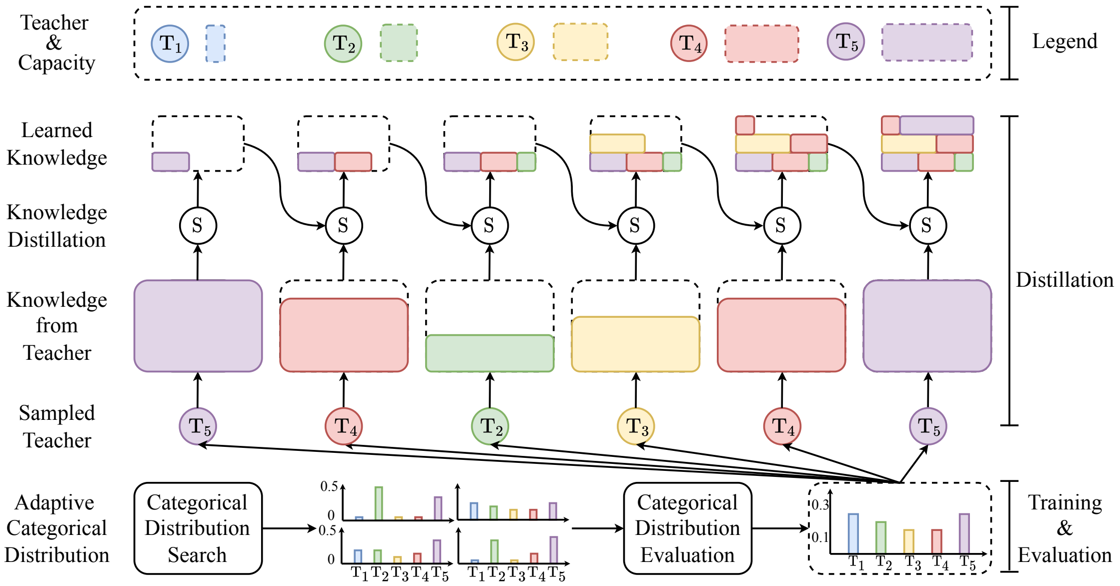

Figure 1: Overview of DynaKD. For various downstream tasks, DynaKD optimizes an adaptive categorical distribution with respect to different-capacity teachers via categorical distribution search, and delivers several categorical distribution candidates which are evaluated to obtain the best one. In each distillation step, a teacher is sampled to transfer its distinct knowledge with whole diversity into student under the adaptive categorical distribution. In entire distillation process, assistant teachers (i.e., T_1 to T_4) contribute to distillation performance improvement via capacity gap alleviation. Best viewed in color.

- We propose a differentiable algorithm to learn the best adaptive categorical distribution via solving a bi-level optimization issue.
- We conduct extensive experiments to verify its effectiveness and generalization ability for both natural language processing and computer vision tasks.

Impact of Quantity and Capacity for Knowledge Distillation

In this paper, similar to SKDBERT, DynaKD employs also five teachers (see Table 1) whose performances can be found in Supplementary Material, as the teacher ensemble for BERT compression. Moreover, we show the distillation performance of the student with respect to each teacher in Table 2. Compared to previous works, e.g., TinyBERT (Jiao et al. 2020), DynaKD employs more and stronger teachers for student distillation. However, this does not mean that the comparison between DynaKD and previous works is unfair.

Quantity: More Teacher is not Always Better

In this section, we employ MT-BERT (Wu, Wu, and Huang 2021) which is a popular multi-teacher distillation paradigm for BERT compression, to verify the impact of teacher quantity on distillation performance. On the one hand, we employ the teacher ensemble used in DynaKD for MT-BERT. On the other hand, similar to DynaKD, we use only the multi-teacher distillation loss, instead of the multi-teacher hidden loss and the task-specific loss for MT-BERT. The experimental results are shown in Table 3. Moreover, we give the implementation details in Supplementary Material.

Model	Name	Layer	Hidden Size	Head	#Params (M)
Student	DynaKD	6	768	12	66.0
Teacher	T_1	8	768	12	81.1
	T_2	10	768	12	95.3
	T_3	12	768	12	110
	T_4	24	1024	16	335
	T_5†	24	1024	16	335

Table 1: The architecture of student and multi-level teachers. † means that the teacher model is pre-trained with whole word masking. The student and all teachers can be downloaded from https://huggingface.co/huawei-noah/TinyBERT_General_6L_768D and https://github.com/google-research/bert, respectively.

Teacher	MRPC	RTE	CoLA	SST-2	QQP	QNLI	MNLI	Avg
	$\frac{F1+acc}{2}$	acc	Mcc	acc	$\frac{F1+acc}{2}$	acc	m	
T_1	89.6	**73.3**	45.8	92.0	88.9	91.1	82.9	80.5
T_2	89.7	71.8	46.2	92.3	89.0	91.3	83.2	80.5
T_3	89.1	71.5	46.9	**93.1**	88.9	**91.4**	82.8	80.5
T_4	**90.0**	72.9	**48.9**	92.1	88.9	91.2	83.4	**81.1**
T_5	89.5	72.6	48.3	92.4	**89.0**	91.3	**83.5**	80.9

Table 2: The distillation performance of student with various teachers on GLUE-dev.

Various teachers contribute to improving the distillation performance via abundant prediction diversities (Tran et al. 2020). However, weighted or average output of multiple teachers is prone to losing the prediction diversity of each characteristic teacher. For instance, multi-teacher KD con-

Task	**MRPC**	**RTE**	**CoLA**	**SST-2**	**QQP**	**QNLI**	**MNLI**
Metrics	$\frac{\text{F1+acc}}{2}$	acc	Mcc	acc	$\frac{\text{F1+acc}}{2}$	acc	m
Best Teacher†	T_4	T_1	T_1	T_3	T_2	T_3	T_5
STKD	**90.0**	73.3	49.3	**93.1**	**89.0**	**91.4**	83.5
MTKD	89.7	**73.7**	**50.1**	92.2	88.6	91.1	**83.6**

Table 3: Performances of KD using single and multiple teachers for the student on the development set of GLUE benchmark. We employ five teachers, i.e. T_1 to T_5 shown in Table 1, for single-teacher distillation and multi-teacher distillation. † means that the best teacher for student distillation on each downstream task as shown in Table 2. STKD and MTKD mean single-teacher KD and multi-teacher KD, respectively.

Student	**Teacher**	**MRPC**	**RTE**	**CoLA**	**SST-2**	**QNLI**	**Avg**
		$\frac{\text{F1+acc}}{2}$	acc	Mcc	acc	acc	
TinyBERT	T_3	87.0	67.9	42.2	92.0	91.2	76.1
	T_5	86.7	70.0	40.9	92.0	90.9	76.1

Table 4: Results of TinyBERT with different-capacity teachers on GLUE-dev. These results are obtained by TinyBERT with the fine-tuned teacher model of DynaKD using the code publicly released by the authors (Jiao et al. 2020).

tributes to only improving the performance on three out of seven downstream tasks. In each distillation step, DynaKD stochastically samples a single teacher to preserve the prediction diversity.

Capacity: Stronger Teacher is not Always Better

To verify the impact of stronger teacher on vanilla distillation paradigm, we employ T_3 which is used in TinyBERT (Jiao et al. 2020) and T_5 which is the strongest teacher used in DynaKD, as the teachers to distill TinyBERT on five downstream tasks for a fair comparison. Following TinyBERT (Jiao et al. 2020), we implement the experiments with batch sizes of {16, 32} and learning rates of {1e-5, 2e-5, 3e-5}, and choose the best result to show in Table 4.

We can observe that the strong teacher T_5 contributes to only improving the performance on RTE. For the above phenomenon, the main reason is that a capacity gap (Mirzadeh et al. 2020) exists between the strongest teacher T_5 and student which is prone to obtaining unsatisfactory performance. To fill the capacity gap, we employ several assistant teachers, i.e., T_1 to T_4 whose capacities lie between the student and the strongest teacher, to transfer the knowledge from T_5 into the student.

The Proposed DynaKD

Overview

With an adaptive categorical distribution $\text{Cat}(\theta)$, similar to SKDBERT (Ding et al. 2023), DynaKD samples a teacher $\hat{\text{T}}$ from a teacher ensemble which consists of n multi-level BERT-style teachers $\text{T}_{1:n}$, to transfer knowledge into student S in each distillation step. The objective function of DynaKD can be expressed as

$$\mathcal{L}(w) = \sum_{x \in \mathcal{X}} \mathcal{L}_d(f_{\hat{\text{T}} \in \text{T}_{1:n}}(x), f_{\text{S}}(x; w)), \tag{1}$$

where $\mathcal{L}_d$ represents distilled loss function to compute the difference between the student S with learnable parameter w and the sampled teacher $\hat{\text{T}}$, $\mathcal{X}$ denotes the training data, $f_{\hat{\text{T}} \in \text{T}_{1:n}}(\cdot)$ and $f_{\text{S}}(\cdot)$ denote the logits from $\hat{\text{T}}$ and S, respectively.

Compare to previous KD paradigms, there are three differences in DynaKD:

1. **Multiple teachers in global view but single teacher in local view for prediction diversity preservation:** In each step (i.e., local view), only the sampled teacher is used to provide prediction for student. In entire procedure (i.e., global view), all teachers are used to transfer knowledge into student.
2. **Multi-level assistant teachers for filling the capacity gap in global view:** In local view, student is directly guided by the sampled teacher whose capacity may be too strong to knowledge transfer. In global view, learned knowledge from assistant teachers contributes to transferring the strongest teacher's knowledge into student.
3. **Adaptive categorical distribution for teacher importance control:** In global view, DynaKD employs an adaptive categorical distribution which can be optimized with respect to various downstream tasks for choosing appropriate teacher in local view.

In DynaKD, $\text{Cat}(\theta)$ where $\theta = \{\theta_{1:n}\}$ and $\sum_{i=1}^{n} \theta_i = 1$, is employed to sample the teacher from the teacher ensemble. Particularly, the probability $p(\text{T}_i)$ of T_i being sampled is θ_i. The best adaptive categorical distribution $\text{Cat}^*(\theta)$ varies with different downstream tasks. For that, DynaKD employs a two-phase paradigm: 1) search and 2) evaluation.

Search Phase

Problem Formulation DynaKD has two groups of learnable parameter: 1) θ of adaptive categorical distribution and 2) w of student. We split original training data into training and validation subsets, and denote $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ as the entropy losses on training and validation subsets, respectively. Both $\mathcal{L}_{train}$ and $\mathcal{L}_{val}$ are determined not only by $\text{Cat}(\theta)$, but also by w. Particularly, DynaKD aims to learn the best categorical distribution $\text{Cat}(\theta^*)$ that minimizes the validation loss $\mathcal{L}_{val}(w^*, \text{Cat}(\theta))$, where the weights w^* associated with the categorical distribution $\text{Cat}(\theta)$ are obtained by $\text{argmin}_w \; \mathcal{L}_{train}(w, \text{Cat}(\theta))$. Consequently, DynaKD can be considered as a bilevel optimization problem (Colson, Marcotte, and Savard 2007) with upper-level variable $\text{Cat}(\theta)$ and lower-level variable w:

$$\begin{aligned} \min_{\text{Cat}(\theta)} \quad & \mathcal{L}_{val}(w^*(\text{Cat}(\theta)), \text{Cat}(\theta)), \\ \text{s.t.} \quad & w^*(\text{Cat}(\theta)) = \underset{w}{\text{argmin}} \; \mathcal{L}_{train}(w, \text{Cat}(\theta)). \end{aligned} \tag{2}$$

We optimize θ of adaptive categorical distribution (Sec.) and w of student (Sec.) in an alternate and iterative way, and show the optimization algorithm in Algorithm 1.

Algorithm 1: Search and Evaluation Phases of DynaKD

```
Output: Initialize categorical distribution $\mathrm{Cat}(\theta)$, weights $w$ of student, candidate number $M$ for evaluation, maximum step $N$, current step $n = 0$
Input: $M$ categorical distribution candidates Computing step number $m$ for saving categorical distribution candidate, i.e., $m = \lfloor \frac{N}{M} \rfloor$
  while $n < N$ do
    Update $w$ by descending Eq. (5)
    Update $\mathrm{Cat}(\theta)$ by descending Eq. (4)
    $n = n + 1$
  end while
  if $n > 0$ and $n \bmod m == 0$ then
    Delivering current $\mathrm{Cat}(\theta)$ as a categorical distribution candidate
  end if
  Evaluate each categorical distribution candidate to choose the best one
```

Categorical Distribution Search For categorical distribution search, w is frozen. We employ Continuous Relaxation (CR) (Liu et al. 2019a) as the technique to optimize $\mathrm{Cat}(\theta)$ in a differentiable way via computing mixture of logits with respect to teachers as

$$\overline{f}_{\mathrm{T}_{1:n}}(x; \mathrm{Cat}(\theta)) = \sum_{i=1}^{n} \theta_i f_{\mathrm{T}_i}(x). \tag{3}$$

Subsequently, the gradient with respect to $\mathrm{Cat}(\theta)$ dubbed $\nabla_{\mathrm{Cat}(\theta)} \mathcal{L}_{val}(w^*(\mathrm{Cat}(\theta)), \mathrm{Cat}(\theta))$ can be computed by an approximation scheme:

$$\nabla_{\mathrm{Cat}(\theta)} \mathcal{L}_{val}(w - \alpha \nabla_w \mathcal{L}_{train}(w, \mathrm{Cat}(\theta)), \mathrm{Cat}(\theta)), \tag{4}$$

where w and α indicate the current weights of the student and the learning rate of categorical distribution, respectively. In particular, we employ w with a single-step adapting (i.e., $w - \alpha \nabla_w \mathcal{L}_{train}(w, \mathrm{Cat}(\theta))$ to appropriate $w^*(\mathrm{Cat}(\theta))$ for avoiding the inner optimization in Eq. (2). This appropriation scheme has been widely used in meta-learning (Finn, Abbeel, and Levine 2017) and Neural Architecture Search (NAS) (Liu et al. 2019a).

Student Distillation For student distillation, $\mathrm{Cat}(\theta)$ is frozen. Similar to Eq. (1), we utilize the following object function:

$$\mathcal{L}(w) = \sum_{x \in \mathcal{X}} \hat{\theta} \mathcal{L}_d(f_{\hat{\mathrm{T}} \in \mathrm{T}_{1:n}}(x), f_{\mathrm{S}}(x; w)), \tag{5}$$

where $\hat{\theta}$ indicates the probability of the teacher $\hat{\mathrm{T}}$ being sampled from $\mathrm{T}_{1:n}$ according to $\mathrm{Cat}(\theta)$.

Evaluation Phase

Generation. Following NAS (Liu et al. 2019a; Ding et al. 2021), search phase delivers M categorical distribution candidates whose generation manner can be found in Algorithm 1, where M candidates are generated every $\lfloor \frac{N}{M} \rfloor$ steps.

Evaluation. Subsequently, we evaluate each candidate to choose the optimal categorical distribution $\mathrm{Cat}^*(\theta)$. We train w of student of DynaKD from scratch with each candidate of $\mathrm{Cat}(\theta)$ and evaluate each candidate on development set. More details can be found in Supplementary Material.

Experiments and Results

Datasets and Settings

Datasets. We evaluate the proposed DynaKD on GLUE benchmark, including MRPC, RTE, CoLA, SST-2, QQP, QNLI and MNLI.

Settings. We employ the development set of GLUE benchmark dubbed as GLUE-dev, for categorical distribution evaluation of DynaKD. We employ a teacher ensemble which consists of 5 BERT-style teachers, to distill a 6-layer BERT-style student dubbed DynaKD. The architecture information of the student and the teachers can be found in Table 1. On the one hand, we employ T_5 as target teacher to transfer abundant knowledge into student for distillation performance improvement. However, there is a large capacity gap between the target teacher and student which is prone to underutilization of target teacher (Mirzadeh et al. 2020). On the other hand, we employ weak T_1 to T_4 (refer to Table 1) as assistant teachers whose capacities are stronger than student but weaker than the strongest teacher (i.e., T_5), to transfer their knowledge into student for filling the above capacity gap between the target teacher and student. Moreover, we obtain all experimental results on NVIDIA A100 GPU with AMD EPYC 7642 48-Core Processor.

DynaKD Optimization

Search Phase

Data Split. The original training set of each downstream task in GLUE benchmark is split fifty-fifty into two subsets, i.e., validation subset for categorical distribution search and training subset for student distillation.

Categorical Distribution Search. We employ Adam as the optimizer for adaptive categorical distribution search. For various tasks, we utilize identical hyper-parameters except learning rate as shown in Supplementary Material.

Student Distillation. For various downstream tasks, we employ other Adam as the optimizer with identical hyper-parameters, except batch size, learning rate and training epoch number, for student distillation. We show the detailed hyper-parameters in Supplementary Material.

Evaluation Phase DynaKD delivers 15 categorical distribution candidates in search phase, and trains the student with each candidate from scratch to choose the optimal categorical distribution. In addition to epoch number, other hyper-parameters (e.g., batch size, learning rate, etc.) are identical to student distillation on various downstream tasks as shown in Supplementary Material. The epoch number is set to 15 on MRPC, RTE, CoLA tasks, and 5 on SST-2, QQP, QNLI and MNLI tasks.

Model	**MRPC**	**RTE**	**CoLA**	**SST-2**	**QQP**	**QNLI**	**MNLI**
	F1/acc	acc	Mcc	acc	F1/acc	acc	m
Poor Man's BERT$_6$ (Sajjad et al. 2020)	-/80.2	65.0	-	90.3	-/90.4	87.6	81.1
DistilBERT$_6$ (Sanh et al. 2019)	87.5/-	59.9	51.3	92.7	-/88.5	89.2	82.2
LayerDrop (Fan, Grave, and Joulin 2020)†	85.9/-	65.2	45.4	90.7	-/88.3	88.4	80.7
BERT-of-Theseus (Xu et al. 2020)	89.0/-	68.2	51.1	91.5	-/89.6	89.5	82.3
MiniLM (Wang et al. 2020)	88.4/-	71.5	49.2	92.0	-/91.0	91.0	84.0
TinyBERT$_6$ (w/o aug) (Jiao et al. 2020)‡	88.4/-	72.2	42.8	91.6	-/90.6	90.5	83.5
MT-BERT (Wu, Wu, and Huang 2021)§	90.8/87.0	72.2	49.1	92.2	87.1/90.4	91.4	83.8
SKDBERT (Ding et al. 2023)	92.1/89.0	75.5	49.1	92.9	87.9/91.0	91.4	84.1
WID$_{55}$ (Wu et al. 2023)	-/88.2	70.4	**61.7**	92.4	-/91.0	90.1	82.9
DynaKD (ours)	**92.2/89.1**	**76.2**	49.8	**93.0**	**88.1/91.1**	**91.5**	**84.2**

Table 5: Results of DynaKD and other popular approaches on GLUE-dev. All comparative approaches have identical architecture, i.e., 6-layer BERT-style language model with 66 million parameters. † and ‡ indicate that the results are cited from (Xu et al. 2020) and (Zuo et al. 2022), respectively. § indicates that the result is obtained by our settings with the distillation loss described in (Wu, Wu, and Huang 2021), and the experimental details can be found in Supplementary Material.

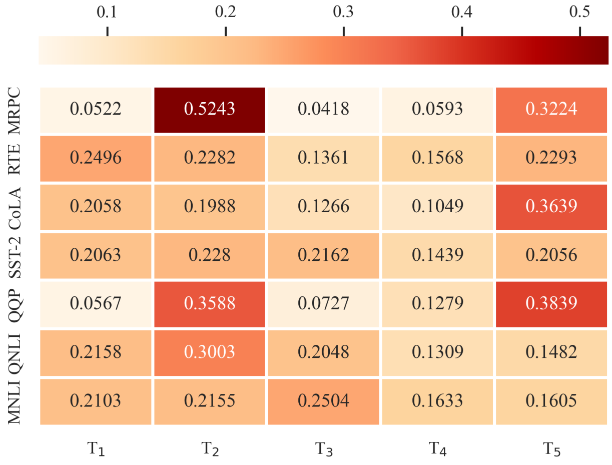

Figure 2: Learned adaptive categorical distributions.

Results and Analysis

Learned Adaptive Categorical Distribution We show the adaptive categorical distributions learned by DynaKD on GLUE benchmark in Figure 2. In the teacher ensemble, each individual teacher shows various importances on different downstream tasks. 1) The strongest teacher T$_5$ plays a dominant role on MRPC, CoLA and QQP tasks rather than all tasks. 2) Assistant teachers provide also useful knowledge to fill the above mentioned capacity gap for student distillation on downstream tasks. For instance, the importance of T$_1$ is the largest on the task of RTE. T$_2$ shows the most important role on MRPC and QNLI tasks. T$_3$ plays the most important role on the task of MNLI.

Performance on GLUE Benchmark Table 5 summarizes the performance of DynaKD and the comparative approaches, e.g., Poor Man's BERT (Sajjad et al. 2020), DistilBERT (Sanh et al. 2019), LayerDrop (Fan, Grave, and Joulin 2020), BERT-PKD (Sun et al. 2019), BERT-of-Theseus (Xu et al. 2020), MiniLM (Wang et al. 2020), TinyBERT (Jiao et al. 2020), MT-BERT (Wu, Wu, and Huang 2021), on GLUE-dev. DynaKD achieves state-of-the-art performance on six out of seven tasks. DynaKD contributes to particularly achieving better performance on those tasks with small data size, e.g., MRPC and RTE. On MRPC, DynaKD achieves 92.2 F1 score and 89.1 accuracy score which are 1.4 and 2.1 point higher than previous state-of-the-art MT-BERT (Wu, Wu, and Huang 2021), respectively. On the other hand, compared to TinyBERT (Jiao et al. 2020) and MT-BERT (Wu, Wu, and Huang 2021) on RTE task, DynaKD achieves 4.0 point absolute improvement.

Ablation Studies

Performance Comparison to Other Knowledge Distillation Paradigms

To verify the effectiveness of the proposed KD paradigm of DynaKD, we compare it with single-teacher KD paradigm and several multi-teacher KD paradigms, e.g., AvgKD (Hinton, Vinyals, and Dean 2015), TAKD (Mirzadeh et al. 2020) via extensive experiments on GLUE benchmark under identical experimental settings. The experimental results are shown in Table 6. Moreover, the experimental settings can be found in Supplementary Material.

For single-teacher KD paradigm, the strongest teacher may not be the best teacher for student distillation. Capacity gap (Mirzadeh et al. 2020) between the strong-capacity teacher and weak-capacity student plays an important role for this phenomenon. For multi-teacher AvgKD, the diversity losing issue leads to worse performance than single-teacher KD paradigm except QQP task, due to using the ensemble of teacher outputs. For multi-teacher TAKD, weak-capacity teachers dramatically reduce the distillation performance of student. In TAKD, the weakest teacher assistant (e.g., T$_1$ of the teacher ensemble) transfers mixture of knowledge which learned from previous stronger teacher assistants (e.g., T$_{02}$ to T$_4$ of the teacher ensemble) into the student. As a result, the performance of TAKD is very sensitive to the capacity of the weakest teacher assistant.

KD Paradigm	**Teacher**	**MRPC**	**RTE**	**CoLA**	**SST-2**	**QQP**	**QNLI**	**MNLI**	**Avg**
		$\frac{F1+acc}{2}$	acc	Mcc	acc	$\frac{F1+acc}{2}$	acc	m	
Single-teacher	T_1	89.6	73.3	49.3	92.0	88.9	91.1	82.9	81.0
	T_2	89.7	71.8	48.5	92.3	89.0	91.3	83.2	80.8
	T_3	89.1	71.5	46.9	**93.1**	88.9	91.4	82.8	80.5
	T_4	90.0	72.9	47.7	92.1	88.9	91.2	83.4	80.9
	T_5	89.5	72.6	48.3	92.4	89.0	91.3	83.5	80.9
AvgKD (Hinton, Vinyals, and Dean 2015)	T_1-T_5	89.9	72.9	48.4	92.2	89.0	91.2	83.4	81.0
TAKD (Mirzadeh et al. 2020)	T_1-T_5	89.3	71.8	47.8	92.7	88.7	91.4	83.4	80.7
DynaKD (ours)	T_1-T_5	**90.7**	**76.2**	**49.8**	93.0	**89.6**	**91.5**	**84.2**	**82.1**

Table 6: Distillation performance of student with various distillation paradigms on GLUE-dev.

Categorical Distribution	**Teacher**	**MRPC**	**RTE**	**CoLA**	**SST-2**	**QQP**	**QNLI**	**MNLI**	**Avg**
		$\frac{F1+acc}{2}$	acc	Mcc	acc	$\frac{F1+acc}{2}$	acc	m	
Uniform (Ding et al. 2023)	T_1-T_5	89.8	73.3	48.8	92.7	89.5	91.4	84.1	81.4
Teacher-rank (Ding et al. 2023)	T_1-T_5	90.5	75.5	49.1	92.1	89.5	91.2	83.9	81.7
Student-rank (Ding et al. 2023)	T_1-T_5	89.9	73.7	47.4	92.9	89.4	91.2	84.0	81.2
Adaptive (ours)	T_1-T_5	**90.7**	**76.2**	**49.8**	**93.0**	**89.6**	**91.5**	**84.2**	**82.1**

Table 7: Distillation performance of student with various hand-crafted categorical distributions on GLUE-dev.

Approach	TinyBERT					DynaKD				
Task	MRPC	RTE	CoLA	SST-2	QNLI	MRPC	RTE	CoLA	SST-2	QNLI
Transformer Layer Distillation	3.42	2.80	12.72	12.90	61.94	0	0	0	0	0
Categorical Distribution Search	0	0	0	0	0	0.08	0.05	0.32	0.64	1.23
Prediction Layer Distillation†	5.04	3.24	4.35	31.71	187.62	1.05	0.75	3.45	3.90	10.95
Total Cost	8.46	6.04	17.07	44.61	249.56	1.13	0.8	3.77	4.54	12.18

Table 8: The computation cost (hours) of TinyBERT and DynaKD on five downstream tasks. These results about TinyBERT are obtained by following the experimental settings described in (Jiao et al. 2020) with the code publicly released by the authors (Jiao et al. 2020). † For TinyBERT, the cost is obtained by 6 groups of experiment with various hyper-parameters (i.e., batch sizes of {16, 32} and learning rates of {1e-5, 2e-5, 3e-5}) on augmentation data. For DynaKD, the cost is obtained by 15 groups of experiment on vanilla data with different categorical distributions learned in the process of categorical distribution search.

Performance Comparison with Hand-crafted Categorical Distributions

In this section, we employ three types of hand-crafted categorical distribution proposed in SKDBERT (Ding et al. 2023) to examine the effectiveness of the adaptive categorical distribution, which can be dynamically optimized according to various tasks.

Three Types of Hand-crafted Categorical Distribution We employ three types of hand-crafted categorical distribution proposed in SKDBERT (Ding et al. 2023), i.e., *uniform*, *teacher-rank* and *student-rank*. *Uniform* categorical distribution sets the weight of each teacher to $\frac{1}{n}$, where n refers to the number of teacher. *Teacher-rank* categorical distribution sets the weight of i-th teacher according to $\frac{s_i^{ft}}{\sum_{j=1}^{n} j}$, where $s_i^{ft} = n - r_i^{ft} + 1$ refers to the fine-tuning performance score of i-th teacher with respect to its rank $r_i^{ft} \in [1, \cdots, n]$. Similarly, *Student-rank* categorical distribution sets the weight of i-th teacher according to $\frac{s_i^{dis}}{\sum_{j=1}^{n} j}$, where $s_i^{dis} = n - r_i^{dis} + 1$ refers to the distillation performance score of student distilled by i-th teacher with respect to its rank $r_i^{dis} \in [1, \cdots, n]$.

Experiments For hand-crafted categorical distribution evaluation, we employ Adam as the optimizer, and choose the best batch size and learning rate from {16, 32} and {1e-5, 2e-5, 3e-5}, respectively. Other hyper-parameters are identical to DynaKD. We show the experimental results in Table 7.

For various downstream tasks, hand-crafted categorical distributions play different roles. Uniform categorical distribution achieves the best performance on the tasks of QQP, QNLI and MNLI. Teacher-rank categorical distribution performs the best on the tasks of MRPC, RTE, and CoLA. Student-rank categorical distribution shows the best performance on SST-2 task. However, the proposed adaptive one outperforms all hand-crafted categorical distributions on all tasks. Particularly, compared to the best hand-crafted one, adaptive categorical distribution achieves 0.7 point absolute improvement on the tasks of RTE and CoLA.

Cost Comparison to TinyBERT

In this section, we show the cost of DynaKD in terms of categorical distribution search and evaluation, and compare our approach to TinyBERT with respect to algorithm cost. Experimental results are shown in Table 8 where on five down-

Student	WRN-16-2	WRN-40-1
Teacher	WRN-40-2	WRN-40-2
Student Accuracy	73.26	71.98
Teacher Accuracy	75.61	75.61
KD (Hinton, Vinyals, and Dean 2015)	74.92	73.54
FitNet (Romero et al. 2015)	73.58	72.24
AT (Zagoruyko and Komodakis 2017)	74.08	72.77
SP (Tung and Mori 2019)	73.83	72.43
CC (Peng et al. 2019)	73.56	72.21
VID (Ahn et al. 2019)	74.11	73.30
RKD (Park et al. 2019)	73.35	72.22
PKT (Passalis and Tefas 2018)	74.54	73.45
AB (Heo et al. 2019)	72.50	72.38
FT (Kim, Park, and Kwak 2018)	73.25	71.59
FSP (Yim et al. 2017)	72.91	-
NST (Huang and Wang 2017)	73.68	72.24
CRD (Tian, Krishnan, and Isola 2020)	75.48	74.14
SKD (Uniform) (Ding et al. 2023)	75.52	74.19
SKD (Teacher-rank) (Ding et al. 2023)	75.42	74.63
SKD (Student-rank) (Ding et al. 2023)	75.37	74.29
DynaKD	**76.04**	**74.72**

Table 9: Test accuracy (%) of the proposed DynaKD and other popular distillation approaches on CIFAR-100. All experimental results are cited from (Tian, Krishnan, and Isola 2020). Average of the last epoch over 5 runs.

stream tasks, the cost of DynaKD is 22.42 hours which is 14.5× less than TinyBERT.

The distillation process of TinyBERT can be divided into two phases: 1) transformer layer distillation on augmentation data and 2) prediction layer distillation on augmentation data. The transformer layer distillation of TinyBERT is time-consuming, e.g., it spends about 62 hours on QNLI. Besides, the prediction layer distillation of TinyBERT is also time-consuming due to using large-scale augmentation data.

Differently, DynaKD consists of categorical distribution search and evaluation (i.e., prediction layer distillation). On the one hand, categorical distribution search is efficient, e.g., 1.23 hours on the task of QNLI, due to the gradient-based optimization method. On the other hand, categorical distribution evaluation is also efficient even choosing the best categorical distribution from 15 candidates.

Generalization for Image Classification

The proposed approach is a general KD paradigm for BERT compression. Consequently, we implement also extensive experiments to verify the effectiveness for image classification on CIFAR-100. Experimental results (see Table 9) show that the proposed KD paradigm can also achieve state-of-the-art performance for computer vision tasks. More details with respect to experimental settings and results can be found in Supplementary Material.

Related Work

Pre-trained Language Model

Based on the transformer-style architecture (Vaswani et al. 2017), BERT (Devlin et al. 2019) achieves state-of-the-art performance on different natural language understanding benchmarks, e.g., GLUE, SQuAD. Subsequently, a great number of variants of BERT are proposed, e.g., XLNet (Yang et al. 2019), ELECTRA (Clark et al. 2020) with new pre-training objectives, RoBERTa (Liu et al. 2019b), T5 (Raffel et al. 2020) with larger pre-training corpus, ConvBERT (Jiang et al. 2020) with various architectures and Synthesizer (Tay et al. 2020) with developed transformer-like block w.r.t. the dot-product self-attention mechanism. Besides, previous pre-trained language models often have several hundred million parameters (e.g. 335 million of $\text{BERT}_{\text{LARGE}}$ (Devlin et al. 2019), even 175 billion of GPT-3 (Brown et al. 2020)) which contribute to delivering amazing performance on downstream tasks while exponentially increasing the difficulty of deployment on resource-constrained device. ALBERT (Lan et al. 2020) adopts parameter sharing strategy to reduce the parameters, and achieves competitive performance.

Knowledge Distillation for BERT-style Language Model Compression

To obtain device-friendly BERT-style language model, many KD-based compression approaches have been proposed. DistilBERT (Sanh et al. 2019) compresses a smaller, faster, cheaper and lighter 6-layer BERT-style language model via learning the soft target probabilities of the teacher in the pre-training stage. In MobileBERT (Sun et al. 2020), an inverted-bottleneck BERT-style language model is pre-trained to transfer knowledge to task-agnostic MobileBERT in a layer-to-layer way. The student in MiniLM (Wang et al. 2020) imitates not only the attention distribution of the teacher, but also the deep self-attention knowledge which reflects the difference between values. In both the pre-training and the fine-tuning phases, TinyBERT (Jiao et al. 2020) learns various knowledge from hidden layer, final layer, embedding and self-attention to achieve high performance. MT-BERT (Wu, Wu, and Huang 2021) employs multiple teachers to achieve better performance than single-teacher KD based approaches on several downstream tasks. Our approach is inspired by SKDBERT (Ding et al. 2023) where a teacher is stochastically sampled from a predefined multi-level teacher ensemble in each step to distill the student following hand-crafted categorical distribution.

Conclusion

This work proposes DynaKD, where an adaptive categorical distribution is optimized for stochastic knowledge distillation (Ding et al. 2023). We observe that the categorical distribution plays an important role for obtaining high-performance DynaKD. Consequently, we propose a differentiable optimization framework to learn the best categorical distribution. Extensive experiments on GLUE benchmark show that the proposed DynaKD achieves state-of-the-art performance compared to popular compression approaches on 6 out of 7 GLUE downstream tasks. Moreover, the proposed KD paradigm can also achieve state-of-the-art performance for image classification on CIFAR-100.

References

Ahn, S.; Hu, S. X.; Damianou, A.; Lawrence, N. D.; and Dai, Z. 2019. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 9163–9171.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 1877–1901.

Clark, K.; Luong, M.-T.; Le, Q. V.; and Manning, C. D. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *8th International Conference on Learning Representations, ICLR*.

Colson, B.; Marcotte, P.; and Savard, G. 2007. An overview of bilevel optimization. *Annals of operations research*, 153(1): 235–256.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, volume 1, 4171–4186.

Ding, Z.; Chen, Y.; Li, N.; Zhao, D.; Sun, Z.; and Chen, C. P. 2021. BNAS: Efficient neural architecture search using broad scalable architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 33(9): 5004–5018.

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.

Dong, Y.; Cordonnier, J.-B.; and Loukas, A. 2021. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, 2793–2803. PMLR.

Fan, A.; Grave, E.; and Joulin, A. 2020. Reducing Transformer Depth on Demand with Structured Dropout. In *8th International Conference on Learning Representations, ICLR*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning (ICML)*, 1126–1135. PMLR.

Gao, J.; Xu, H.; Shi, H.; Ren, X.; Philip, L.; Liang, X.; Jiang, X.; and Li, Z. 2022. AutoBERT-zero: Evolving BERT backbone from scratch. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 10663–10671.

Gordon, M. A.; 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. Association for Computational Linguistics.

Heo, B.; Lee, M.; Yun, S.; and Choi, J. Y. 2019. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3779–3787.

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

Huang, Z.; and Wang, N. 2017. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*.

Jiang, Z.-H.; Yu, W.; Zhou, D.; Chen, Y.; Feng, J.; and Yan, S. 2020. Convbert: Improving bert with span-based dynamic convolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 12837–12848.

Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; and Liu, Q. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 4163–4174.

Kim, J.; Park, S.; and Kwak, N. 2018. Paraphrasing complex network: Network compression via factor transfer. In *Advances in Neural Information Processing Systems, NeurIPS*, volume 31.

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

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *8th International Conference on Learning Representations, ICLR*.

Liu, H.; Simonyan, K.; Yang, Y.; et al. 2019a. DARTS: Differentiable architecture search. In *International Conference on Learning Representations (ICLR)*.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mirzadeh, S. I.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; and Ghasemzadeh, H. 2020. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence, AAAI*, volume 34, 5191–5198.

Pan, H.; Wang, C.; Qiu, M.; Zhang, Y.; Li, Y.; and Huang, J. 2021. Meta-KD: A Meta Knowledge Distillation Framework for Language Model Compression across Domains. 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*, 3026–3036.

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR*, 3967–3976.

Passalis, N.; and Tefas, A. 2018. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision, ECCV*, 268–284.

Peng, B.; Jin, X.; Liu, J.; Li, D.; Wu, Y.; Liu, Y.; Zhou, S.; and Zhang, Z. 2019. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV*, 5007–5016.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21: 140:1–140:67.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2015. Fitnets: Hints for thin deep nets. In *3th International Conference on Learning Representations, ICLR*.

Sajjad, H.; Dalvi, F.; Durrani, N.; and Nakov, P. 2020. Poor man's bert: Smaller and faster transformer models. *arXiv preprint arXiv:2004.03844*.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Shen, S.; Dong, Z.; Ye, J.; Ma, L.; Yao, Z.; Gholami, A.; Mahoney, M. W.; and Keutzer, K. 2020. Q-BERT: Hessian based ultra low precision quantization of BERT. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 8815–8821.

Sun, S.; Cheng, Y.; Gan, Z.; and Liu, J. 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*, 4322–4331.

Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; and Zhou, D. 2020. MobileBERT: A Compact Task-Agnostic BERT for Resource-Limited Devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL*, 2158–2170.

Tay, Y.; Bahri, D.; Metzler, D.; Juan, D.; Zhao, Z.; and Zheng, C. 2020. Synthesizer: Rethinking self-attention in transformer models. arXiv 2020. *arXiv preprint arXiv:2005.00743*, 2.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive representation distillation. In *8th International Conference on Learning Representations, ICLR*.

Tran, L.; Veeling, B. S.; Roth, K.; Swiatkowski, J.; Dillon, J. V.; Snoek, J.; Mandt, S.; Salimans, T.; Nowozin, S.; and Jenatton, R. 2020. Hydra: Preserving ensemble diversity for model distillation. In *International Conference on Machine Learning Workshop on Uncertainty and Robustness in Deep Learning*.

Tung, F.; and Mori, G. 2019. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision, ICCV*, 1365–1374.

Turc, I.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *arXiv preprint arXiv:1908.08962*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. volume 30.

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. In *Advances in Neural Information Processing Systems, NeurIPS*, volume 33, 5776–5788.

Wu, C.; Wu, F.; and Huang, Y. 2021. One teacher is enough? pre-trained language model distillation from multiple teachers. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP*, 4408–4413.

Wu, T.; Hou, C.; Zhao, Z.; Lao, S.; Li, J.; Wong, N.; and Yang, Y. 2023. Weight-Inherited Distillation for Task-Agnostic BERT Compression. *arXiv preprint arXiv:2305.09098*.

Xu, C.; Zhou, W.; Ge, T.; Wei, F.; and Zhou, M. 2020. BERT-of-Theseus: Compressing BERT by Progressive Module Replacing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP*, 7859–7869.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems, NeurIPS*, volume 32.

Yim, J.; Joo, D.; Bae, J.; and Kim, J. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 4133–4141.

Zagoruyko, S.; and Komodakis, N. 2017. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer.

Zuo, S.; Zhang, Q.; Liang, C.; He, P.; Zhao, T.; and Chen, W. 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*, 1610–1623.