AUTO DISC: Automatic Distillation Schedule for Large Language Model Compression

Chen Zhang*, Yang Yang†, Qifan Wang*, Jiahao Liu*, Jingang Wang*, Wei Wu†, Dawei Song*
*Beijing Institute of Technology
†Meituan NLP
{czhang,dwsong}@bit.edu.cn
{yangyang113,liujiahao12,wangjingang02,wuwei30}@meituan.com
‡Meta AI
wqfcr@fb.com

Abstract

Driven by the teacher-student paradigm, knowledge distillation is one of the de facto ways for language model compression. Recent studies have uncovered that conventional distillation is less effective when facing a large capacity gap between the teacher and the student, and introduced teacher assistant-based distillation to bridge the gap. As a connection, the scale and the performance of the teacher assistant is crucial for transferring the knowledge from the teacher to the student. However, existing teacher assistant-based methods manually select the scale of the teacher assistant, which fails to identify the teacher assistant with the optimal scale-performance tradeoff. To this end, we propose an Automatic Distillation Schedule (AUTO DISC) for large language model compression. In particular, AUTO DISC first specifies a set of teacher assistant candidates at different scales with gridding and pruning, and then optimizes all candidates in an once-for-all optimization with two approximations. The best teacher assistant scale is automatically selected according to the scale-performance tradeoff. AUTO DISC is evaluated with an extensive set of experiments on a language understanding benchmark GLUE. Experimental results demonstrate the improved performance and applicability of our AUTO DISC. We further apply AUTO DISC on a language model with over one billion parameters and show the scalability of AUTO DISC.

1 Introduction

Language models (LMs) [1–5] achieve promising results in various downstream tasks [6–7], but are inapplicable to those requiring limited budget or low latency [8]. To fulfill the computational requirement, language models can be compressed via a range of strategies such as model quantization [9–10], pruning [11–12], etc., among which knowledge distillation [13–14] is an appealing choice under the teacher-student paradigm. In knowledge distillation, language models serve as teachers and are distilled to small-scale students.

Recent advances [15] have shown that conventional distillation suffers from severe performance decline when facing a large capacity gap between the teacher and the student. To alleviate the shortcoming, teacher assistant-based distillation [16] has been proposed, where the teacher is first distilled into an intermediate-scale teacher assistant. This teacher assistant then serves as an alternative teacher to transfer the knowledge to the student. While teacher assistant-based distillation generally

Preprint. Under review.
Our main contributions are summarized as follows:

- We investigate the impact of teacher assistants with different scales on the performance of the student. Based on the observation, we introduce a new scale-performance tradeoff measure on the teacher assistant that is positively correlated with the student performance.

- We propose to leverage gridding and pruning to specify teacher assistant candidates, and show two properties of the generated candidates. These properties lead to a novel optimization framework that jointly achieves optimizing teacher assistants at every scale in one run.

- We conduct comprehensive experiments on various datasets to validate the effectiveness of the AUTO DISC. Our results on a language model with over one billion parameters show the scalability of our approach. To our best knowledge, our work is the first one exploring the distillation of true large-scale LMs.
2 Methodology

2.1 Problem Formulation

In this section, we formally define the problem of AUTO_DISC. Assuming we have a teacher denoted as \((T, s_t, m_t)\) that should be distilled to a student \((S, s_s, m_s)\), the goal is to find a teacher assistant \((\mathcal{A}, s_a, m_a)\) such that the student performance can be maximized when distilling the teacher to the student via the teacher assistant (i.e., \(T \rightarrow \mathcal{A} \rightarrow S\)). Here the first, second and third elements in a tuple denote the model structure/architecture, the scale, and the performance respectively. It is straightforward that the scale and the performance of the teacher assistant are bounded by the teacher and the student, i.e., \(s_s \leq s_a \leq s_t\) and \(m_s \leq m_a \leq m_t\). Ideally, we are seeking a teacher assistant with small scale and high performance so that the best student can be realized. To achieve an optimal scale-performance balance, we first introduce a new tradeoff metric below:

**Definition 1 (\(\lambda\)-Tradeoff)** The \(\lambda\) scale-performance tradeoff measure of a teacher assistant \((\mathcal{A}, s_a, m_a)\) is defined as

\[
t_a = \frac{m_a}{m_t} + \lambda \cdot (1 - \frac{s_a}{s_t}), \quad \text{where } \lambda \in [0, 1].
\]

It is clear that the value of \(\lambda\)-Tradeoff is bounded by \(1 + \lambda \cdot (1 - \frac{s_s}{s_t})\) when the teacher assistant can achieve the performance of the teacher \((m_a = m_t)\) with the student scale \((s_a = s_s)\). However, in practice, this is impossible as smaller models usually lead to lower performance as shown by the blue curves in Figure 1. We further observe that the \(\lambda\)-Tradeoff (red curves) of the teacher assistant is positively correlated with the performance of the student (green curves). Therefore, the problem can be reformulated as finding an optimal teacher assistant with largest value of \(\lambda\)-Tradeoff:

\[
(\mathcal{A}^*, s^*_a, m^*_a) = \arg\max_{\mathcal{A}, s_a, m_a} t_a = \arg\max_{s_a} \left( \arg\max_{m_a} \left( \arg\max_{\mathcal{A}} t_a \right) \right) \tag{1}
\]

Based on the above reformulation, our methodology can be decomposed into three main steps: specification, optimization, and selection. Essentially, during specification, a set of teacher assistant candidates are generated at different scales. Then the best performing teacher assistant at each scale is obtained through an one-run optimization. Finally, the optimal teacher assistant \(\mathcal{A}^*\) is selected with a linear scan of all scales during selection. After the discovery of the optimal teacher assistant, the teacher assistant can subsequently be distilled to the expected student. An overview of the methodology is given in Figure 2.

2.2 Specification

**Gridding.** Theoretically, one needs to generate candidates at every possible scale to find the optimal solution. However, it is impossible to enumerate all possibilities in a continuous space. Therefore, we discretize the candidate scales into \(n\) discrete values, \(\{\mathcal{A} = (A_k, s_{ak}, m_{ak}) \mid \Delta s_a = (s_t - s_s)/n\}\), with equal slicing between the teacher scale and student scale.

**Pruning.** For candidates at each scale, there are still an infinite number of possible structures, e.g., different combinations of width and depth. A number of approaches have been proposed to identify a good structure at a scale, including dynamic search [12], layer dropping [21] and pruning [11].
In this work, we adopt pruning to assign structures \( \mathcal{A}_k \) to the candidates due to its known advantages in knowledge distillation \([24]\). Concretely, following previous work \([11]\), the pruning starts with the least important parameters/features based on their importance scores, which are approximated by masking the parameterized structures. The technical details of our pruning are supplied in Appendix \([A]\).

Essentially, gridding positions the scales of candidates between the scales of the teacher and student with equal intervals and pruning assigns candidates with pruned structures.

### 2.3 Optimization

A straightforward solution to unearth the performance of each candidate is exhaustively measuring the performance of each by actual distillation, e.g., MANDISC \([15]\). However, the memory footprints and the computational costs apparently can be extremely large with the increasing of the granularity of the candidate scale. To reduce the memory overhead and the computation complexity, we introduce two effective approximations, parameter-sharing and sandwich-optimization, so that the performance of all candidates at different scales can be optimized in an once-for-all optimization. The feasibility of the approximations are guarded by the following two Lemmas.

**Lemma 1 (Incremental Property)** For two candidates \( \mathcal{A}_i \) and \( \mathcal{A}_j \) in the teacher assistant candidate set \( \mathcal{A} \), if \( s_i < s_j \), then we have \( \mathcal{A}_i \subset \mathcal{A}_j \).

This Lemma is an outcome of the pruning approach, which essentially tells that among all candidates obtained from the specification, the structure of a candidate at a smaller scale is a subset of the structure for a candidate at a larger scale. The proof of the Lemma can be found in \([18][19]\).

**Remark 1** The incremental property affirms that a larger-scale candidate can result in a smaller-scale one by continuously pruning less significant parameters, which enables these candidates to be assembled into one sandwich-like model in a parameter-sharing fashion. The memory scale of the sandwich-like model is exactly that of the largest-scale candidate.

**Lemma 2 (Sandwich Rule)** For two candidates \( \mathcal{A}_i \) and \( \mathcal{A}_j \) from candidate set \( \mathcal{A} \), if \( s_i < s_j \), then we have \( m_s \leq m_i \leq m_j \leq m_s \).

The sandwich rule \([23][24]\) states that the performance of a candidate is bounded by the best performance of a larger-scale candidate and smaller-scale, due to the subset structure. Therefore, a candidate can be optimized by alternatively distilling its larger-scale and smaller-scale candidates, without direct distillation.

**Remark 2** The sandwich rule allows us to sub-sample \( \eta \) out of all \( n \ (\eta \leq n) \) filling-like candidates and conduct sandwich-optimization over the sampled candidates, which substantially reduces the computational cost.

With the two approximations, we reduce the memory footprints of all candidates to a distinguished one with the largest-scale. The computational costs are also largely reduced with the sub-sampling. Finally, we formulate the distillation objectives for task-specific distillation (TSD) and task-agnostic distillation (TAD) respectively as below:

\[
\mathcal{L}_{TSD} = \sum_{i=1}^{\eta} \left( CE(Y_T, Y_{A_i}) + \sum_{u=1}^{l} \text{MSE}(X^u_T, X^u_{A_i}) \right) \\
\mathcal{L}_{TAD} = \sum_{i=1}^{\eta} \sum_{j=1}^{h} \left( \text{KL}(Q^j_{R^i_T}, Q^j_{A_{i}}) + \text{KL}(K^j_{R^i_T}, K^j_{A_{i}}) + \text{KL}(V^j_{R^i_T}, V^j_{A_{i}}) \right) \tag{2}
\]

where \( \text{MSE} \), \( \text{CE} \) and \( \text{KL} \) stands for mean squared error, cross entropy and kullback-leibler divergence respectively. \( X^u \) is the intermediate output of the \( u \)-th layer within totally \( l \) layers, \( Y \) is the final prediction. As is taken from MiniLMv2 \([25]\), \( Q^j \) is the query relation matrix of the \( j \)-th head within totally \( h \) attention heads from the last layer, likewise \( K^j \) and \( V^j \) are the key and value relation matrices. \( T \) denotes the teacher and \( A_i \) denotes the \( i \)-th teacher assistant candidate from the sampled pool. The teacher assistants with the best performance at different scales can be obtained after the above sandwich-optimization. The unsampled teacher assistants can be retrieved based on the larger-scale teacher assistant from the sampled pool using the shared parameters.
2.4 Selection

The intuition for imposing the teacher assistant is to adequately sacrifice teacher scale for student performance, where the adequacy should be assured by retaining teacher assistant performance as much as possible. In other words, a good teacher assistant is the one at a small scale yet with nice performance. As observed in the pilot study, the \( \lambda \)-Tradeoff measure is positively correlated with the final student performance and thus is directly used as the selection criterion. The optimal teacher assistant can be identified by selecting the candidate with the best tradeoff measure. The optimal teacher assistant is then distilled to the expected student again following above distillation objectives. Note that the tradeoff measure is also dependent on \( \lambda \). However, we empirically find that the optimal solution of AUTO\( \text{DISC} \) is relatively stable with a wide range of \( \lambda \), and we fix \( \lambda \) to 0.2 in all our experiments. More discussion of the impact of \( \lambda \) is provided in the experiments.

3 Experiments

3.1 Setup

Datasets and Metrics We conduct experiments on a language understanding benchmark GLUE \([6]\). The GLUE originally consists of two sequence classification tasks, SST-2 \([26]\) and CoLA \([27]\), with seven sequence-pair classification tasks, i.e., MRPC \([28]\), STS-B \([29]\), QQP, MNLI \([30]\), QNLI \([31]\), RTE \([32]\) and WNLI \([33]\). We exclude WNLI and CoLA due to the evaluation inconsistency (in other words, compressed LMs get dramatically worse results while original LMs get much better ones as found out in \([22]\)) and use the other seven tasks for evaluation. Following the work in BERT \([1]\), we report F1 on MRPC and QQP, Spearman Correlation scores (Sp Corr) on STS-B, and Accuracy (Acc) on other tasks. Macro average scores (Average) over these seven tasks are computed for overall performance. Results on development sets are reported. We also adopt Wikipedia for pretraining in task-agnostic distillation. The detailed statistics, maximum sequence lengths, and metrics of GLUE and Wikipedia are supplied in Appendix B.

Implementation Details Experiments are carried out on BERT\(_{\text{base}}\) \([1]\) and EncT5\(_{1.5b}\) \([20]\). EncT5 is a language model which achieves competitive performance as T5\(_{3b}\) \([5]\) on GLUE with a nearly encoder-only T5 (incorporated with a decoder layer). Our task-specific experiments are carried out on one Nvidia A100, and \( \eta \) is set to 9 according to our empirical investigation. On the other hand, the task-agnostic experiments are carried out on eight Nvidia A100s with BERT\(_{\text{base}}\) only.\(^1\) The pre-training is armed with a dynamic masking strategy \([21]\). \( \eta \) is set to 1 to substantially reduce computational burden. The number of relation heads is set to 32 since we use deep relation distillation as the task-agnostic distillation objective. Other implementation details are supplied in Appendix C. Generally, the sampling is performed from candidates at scales \{100\%, 95\%, 90\%, ..., 10\%, 5\%\}.

Baselines We compare our model with several state-of-the-art baselines. BERT\(_{L}\), MiniLMv2\(_{L/H}\), and TinyBERT\(_{L/H}\) denote methods via dropping layers and hidden dimensions, while DynaBERT\(_{L/G}\), BERT\(_{L/G}\), and EncT5\(_{L/G}\) represent structured pruning with either local ranking or our global ranking (see Appendix A).

- **Conventional Distillation:** FT \([18]\) indicates direct fine-tuning after pruning. KD \([34]\), PKD \([13]\) and CKD \([35]\) are methods with different objectives, i.e., KD directly distills logits, PKD distills both logits and hidden states and CKD distills token and layer relations. DynaBERT \([12]\) uses structured pruning with a local ranking in each layer. MiniLMv2 \([25]\) is distilled with the deep relation alignment. TinyBERT \([36]\) is distilled with a combination of various feature distillations.

- **Teacher Assistant Distillation:** MAN\( \text{DISC} \) \([15]\) manually selects the best teacher assistant. TA \([14]\) is specifically incorporated with MiniLMv2 for task-agnostic distillation.

3.2 Main Results

Results on Task-specific Distillation Table\([1]\) presents the comparison results of different methods on task-specific distillation at four student scales. The highlighted rows are the results from this work.

\(^1\)We are not able to make it work for EncT5\(_{1.5b}\) and there is no existing guidance as of now.
Table 1: The results of task-specific distillation upon BERT\textsubscript{base}. The best and second best results are boldfaced and underlined.

| Model          | FLOPs | SST-2 | MRPC | STS-B | QQP | MNLI-m/mm | QNLI | RTE | Average |
|----------------|-------|-------|------|-------|-----|-----------|------|-----|---------|
| BERT\textsubscript{base} | 10.9G | 93.8  | 91.5 | 87.1  | 88.4 | 84.9/84.9 | 91.9 | 71.5 | 86.7    |
| BERT\textsubscript{4L-KD} [34] | 3.6G  | 89.6  | 86.9 | 86.4  | 86.1 | 77.7/77.7 | 85.1 | 65.3 | 81.9    |
| BERT\textsubscript{4L-PKD} [13] | 3.6G  | 89.9  | 87.6 | 86.4  | 86.0 | 77.7/77.7 | 85.0 | 65.3 | 82.0    |
| BERT\textsubscript{4L-CRD} [53] | 3.6G  | 86.6  | 87.2 | 86.4  | 86.2 | 77.7/77.9 | 85.0 | 64.6 | 81.8    |
| DynaBERT\textsubscript{15%} [12] | 3.3G  | 90.3  | 87.4 | 87.2  | 86.6 | 81.5/81.8 | 89.1 | 66.1 | 83.7    |
| BERT\textsubscript{3L-FT} [13] | 3.3G  | 91.9  | 88.5 | 87.2  | 87.7 | 82.0/82.6 | 89.5 | 69.0 | 84.8    |
| BERT\textsubscript{3L-KD} [34] | 3.3G  | 92.0  | 88.9 | 86.8  | 87.8 | 82.2/82.7 | 89.9 | 68.2 | 84.8    |
| BERT\textsubscript{3L-L\textsubscript{TSD}} | 3.3G  | 91.9  | 89.5 | 86.4  | 88.0 | 82.5/82.8 | 89.9 | 68.6 | 84.9    |
| BERT\textsubscript{2L-KD} [34] | 1.8G  | 86.8  | 82.5 | 46.8  | 83.7 | 73.5/73.1 | 79.6 | 58.1 | 73.0    |
| BERT\textsubscript{2L-PKD} [13] | 1.8G  | 86.7  | 82.4 | 46.8  | 83.7 | 73.4/73.0 | 79.7 | 57.4 | 72.9    |
| BERT\textsubscript{2L-CRD} [53] | 1.8G  | 86.4  | 82.3 | 48.6  | 83.6 | 73.3/73.0 | 79.1 | 56.7 | 72.9    |
| DynaBERT\textsubscript{15%} [12] | 2.2G  | 89.1  | 85.1 | 84.7  | 84.3 | 78.3/79.0 | 86.6 | 61.4 | 81.1    |
| BERT\textsubscript{15%}-FT [13] | 1.6G  | 89.9  | 87.1 | 85.6  | 86.1 | 79.9/80.1 | 85.7 | 63.9 | 82.3    |
| BERT\textsubscript{15%}-KD [34] | 1.6G  | 89.9  | 88.6 | 85.1  | 86.2 | 79.8/80.2 | 85.6 | 63.9 | 82.4    |
| BERT\textsubscript{15%}-L\textsubscript{TSD} | 1.6G  | 90.1  | 88.9 | 85.1  | 86.5 | 80.0/80.2 | 86.0 | 65.3 | 82.8    |
| w/ M\textsubscript{ANDISC} [15] | 1.6G  | 89.8  | 87.5 | 85.4  | 86.9 | 81.0/80.1 | 86.1 | 68.2 | 83.2    |
| w/ AUTO\textsubscript{DISC} | 1.6G  | 89.8  | 88.2 | 85.8  | 86.6 | 80.3/79.9 | 87.3 | 68.2 | 83.3    |
| BERT\textsubscript{1L}-FT [13] | 1.1G  | 88.2  | 84.8 | 84.7  | 84.4 | 77.6/77.3 | 84.3 | 65.3 | 80.8    |
| BERT\textsubscript{1L-KD} [34] | 1.1G  | 88.2  | 87.6 | 84.0  | 84.4 | 77.6/77.4 | 84.3 | 67.2 | 81.3    |
| BERT\textsubscript{1L-CRD} [53] | 1.1G  | 88.2  | 87.4 | 84.0  | 84.6 | 77.7/77.5 | 84.9 | 66.8 | 81.5    |
| w/ M\textsubscript{ANDISC} [15] | 1.1G  | 89.0  | 88.2 | 84.8  | 84.8 | 78.3/77.8 | 85.3 | 66.8 | 81.9    |
| w/ AUTO\textsubscript{DISC} | 1.1G  | 89.1  | 88.4 | 85.4  | 84.9 | 78.2/78.6 | 86.3 | 68.2 | 82.4    |
| BERT\textsubscript{15%}-FT [13] | 0.5G  | 85.4  | 82.8 | 84.1  | 82.6 | 72.5/73.3 | 81.7 | 63.9 | 78.3    |
| BERT\textsubscript{15%}-KD [34] | 0.5G  | 85.6  | 84.0 | 83.8  | 82.5 | 72.6/73.2 | 81.6 | 63.2 | 78.3    |
| BERT\textsubscript{15%}-L\textsubscript{TSD} | 0.5G  | 85.4  | 85.5 | 83.9  | 82.7 | 73.0/73.4 | 82.7 | 63.2 | 78.7    |
| w/ M\textsubscript{ANDISC} [15] | 0.5G  | 86.1  | 87.0 | 84.1  | 83.8 | 73.7/73.6 | 82.9 | 65.7 | 79.6    |
| w/ AUTO\textsubscript{DISC} | 0.5G  | 86.9  | 87.6 | 84.8  | 83.5 | 72.7/74.5 | 84.0 | 66.8 | 80.1    |

where \( \mathcal{L}_{TSD} \) denotes the student directly obtained from distillation and AUTO\textsubscript{DISC} denotes the student with additional distillation using the optimal teacher assistant. There are several observations: First, \( \mathcal{L}_{TSD} \) achieves the best performance among all conventional distillation methods, while AUTO\textsubscript{DISC} further improves the model and obtains similar or even better results compared to M\textsubscript{ANDISC}. This validates the improved performance of AUTO\textsubscript{DISC} for automatically identifying a good teacher assistant. Notably, for further smaller-scale BERT\textsubscript{15%}, the improvement still holds, as supplied in Appendix [D]. Second, conventional distillations generate reasonable results at large student scale 30% but fail to maintain the student performance at small scale 15% (with extremely worse results at scales 10% and 5% which are not included). Nonetheless, AUTO\textsubscript{DISC} consistently outperforms the baselines at all scales. Additional comparisons of practical inference measurement are supplied in Appendix [E]. Third, pruning based models perform much better compared to the layer dropping methods, e.g., BERT\textsubscript{15%}-KD achieves much higher score than FLOPs-matched BERT\textsubscript{2L}-KD, which verifies the effectiveness of pruning approach in knowledge distillation. Moreover, we discover the global ranking strategy surpasses the local ranking one by comparing BERT\textsubscript{15%}-L\textsubscript{TSD} to FLOPs-matched DynaBERT\textsubscript{15%}. We speculate the narrow structures induced by the local ranking strategy are not effective for small-scale students. The distributions of example pruned structures are supplied in Appendix [F].

Results on Task-agnostic Distillation We also apply AUTO\textsubscript{DISC} to task-agnostic distillation and report the results in Table [2]. The first glimpse is that \( \mathcal{L}_{TAD} \) surpasses \( \mathcal{L}_{TSD} \), indicating the deep relation alignment is more suitable for task-agnostic distillation. Surprisingly, we discover that the pruned structures can boost the performance of MiniLMv2 and establish a new state-of-the-art for conventional task-agnostic distillation. Another interesting observation is that teacher assistant-based distillations do not improve the performance over conventional distillations until the scale is reduced to 5%, indicating that conventional distillations are already promising choices on task-agnostic distillations. Nonetheless, we still argue the applicability of AUTO\textsubscript{DISC} to task-agnostic distillation for a performance guarantee. Note that TinyBERT is less effective without data augmentation, and its results with data augmentation are supplied in Appendix [G].
Table 2: The results of task-agnostic distillation upon BERTbase. The best and second best results are boldfaced and underlined. TA stands for teacher assistant. The results of TinyBERT are reproduced based on their released checkpoints without data augmentation for a fair comparison.

| Model | FLOPs | SST-2 | MRPC | STS-B | QQP | MNLI-m/mm | QNLI | RTE | Average |
|-------|-------|-------|------|-------|-----|------------|------|------|---------|
| BERTbase | 10.9G | 93.8 | 91.5 | 87.1 | 88.4 | 84.9/84.9 | 91.9 | 71.5 | 86.7 |
| MiniLMv2L3B4H | 0.9G | 90.0 | 88.6 | 87.2 | 86.1 | 80.0/80.3 | 87.9 | 67.2 | 83.4 |
| w/ TA [14] | 0.9G | 90.0 | 88.5 | 87.3 | 86.3 | 80.1/80.7 | 88.0 | 66.4 | 83.4 |
| BERTbase-FT [18] | 1.1G | 84.6 | 83.1 | 83.8 | 84.5 | 75.3/75.4 | 83.2 | 56.7 | 78.3 |
| BERTbase-LTSD [18] | 1.1G | 90.7 | 89.0 | 87.0 | 85.9 | 78.4/78.2 | 86.0 | 66.4 | 82.7 |
| BERTbase-LTAD | 1.1G | 92.0 | 90.1 | 87.9 | 86.6 | 80.0/80.3 | 88.0 | 67.2 | 84.0 |
| w/ MANDISC [15] | 1.1G | 91.5 | 90.3 | 87.8 | 86.6 | 80.0/80.1 | 88.6 | 67.2 | 84.0 |
| w/ AUTODISC | 1.1G | 91.4 | 90.0 | 87.5 | 86.6 | 79.8/80.0 | 88.0 | 67.2 | 83.8 |
| TinyBERT_L3B4H | 0.6G | 88.3 | 88.5 | 84.3 | 84.0 | 77.0/77.4 | 82.5 | 63.5 | 80.7 |
| w/ TA [14] | 0.7G | 89.1 | 89.1 | 86.6 | 85.4 | 77.8/78.4 | 87.2 | 66.1 | 82.5 |
| BERT55-FT [18] | 0.5G | 84.1 | 82.4 | 81.8 | 83.7 | 74.4/74.9 | 82.5 | 57.0 | 77.6 |
| BERT55-LTAD | 0.5G | 90.9 | 89.4 | 87.7 | 85.8 | 79.2/79.8 | 87.3 | 65.7 | 83.2 |
| w/ MANDISC [15] | 0.5G | 90.1 | 89.7 | 87.4 | 85.6 | 79.3/79.7 | 87.1 | 67.9 | 83.4 |
| w/ AUTODISC | 0.5G | 89.3 | 89.7 | 87.4 | 85.9 | 79.2/79.4 | 86.9 | 69.7 | 83.4 |

Table 3: The results of task-specific distillation upon EncT51.5b. The best and second best results are boldfaced and underlined.

| Model | FLOPs | SST-2 | MRPC | STS-B | QQP | MNLI-m/mm | QNLI | RTE | Average |
|-------|-------|-------|------|-------|-----|------------|------|------|---------|
| EncT51.5b | 155.9G | 96.9 | 95.1 | 92.3 | 90.0 | 90.7/90.9 | 95.0 | 88.5 | 92.4 |
| EncT51.5b-FT [18] | 15.6G | 91.6 | 87.1 | 86.7 | 87.9 | 81.9/87.0 | 66.1 | 91.6 | 83.8 |
| EncT51.5b-KD [34] | 15.6G | 92.2 | 86.6 | 86.6 | 87.9 | 83.6/83.8 | 88.1 | 63.5 | 84.1 |
| EncT51.5b-LTSD | 15.6G | 94.5 | 90.2 | 87.4 | 87.9 | 84.7/84.1 | 90.8 | 67.5 | 85.9 |
| w/ MANDISC [15] | 15.6G | 94.6 | 90.5 | 88.0 | 88.1 | 86.2/85.1 | 91.5 | 70.4 | 86.8 |
| w/ AUTODISC | 15.6G | 94.6 | 91.5 | 87.8 | 87.3 | 85.9/85.0 | 91.1 | 72.2 | 86.9 |
| EncT53.5b-FT [18] | 7.8G | 90.1 | 84.8 | 84.7 | 86.5 | 78.0/78.2 | 83.9 | 62.8 | 81.1 |
| EncT53.5b-KD [34] | 7.8G | 89.9 | 85.1 | 85.4 | 86.6 | 79.4/79.6 | 84.2 | 55.6 | 80.7 |
| EncT53.5b-LTSD | 7.8G | 92.9 | 88.0 | 83.4 | 85.4 | 79.6/80.0 | 87.0 | 58.8 | 81.9 |
| w/ MANDISC [15] | 7.8G | 93.0 | 88.0 | 83.9 | 86.5 | 81.2/81.6 | 88.1 | 67.5 | 83.7 |
| w/ AUTODISC | 7.8G | 93.8 | 89.8 | 85.3 | 86.7 | 82.9/82.7 | 89.2 | 64.6 | 84.4 |

Results on Large-scale Distillation As is shown in Table 3, we conduct a similar comparison on a large-scale language model, EncT51.5b, with over one billion parameters. The very first results of a large-scale LM also exhibit an akin trend as the one in BERTbase. And the results for a moderate-scale LM BERTlarge are supplied in Appendix I. We therefore conclude that the scalability of AUTODISC is also compelling. Reversely, the results of AUTODISC on small-scale LMs are supplied in Appendix I.

3.3 Ablation Study and Analysis

Scale-performance Tradeoff To validate the existence of scale-performance tradeoff, we use teacher assistants at different scales for MANDISC and plot performance variations of these schedules upon BERTbase in Figure 3(a). It can be seen that reducing the scale can lead to performance improvement until a certain scale, after which performance degradation is witnessed. Almost all manual schedules underperform the automatic one. We attribute the inferiority of manual schedules to improper scale-performance tradeoffs, as concentrating only on either scale or performance will give rise to a trivial solution with pareto optimality [37, 38]. The overall phenomenon implies the existence of scale-performance tradeoff. Similar phenomenon is also observed in EncT5, which is supplied in Appendix I. One may note that AUTODISC does not achieve the best performance, which is expected as MANDISC exhaustively searches for the teacher assistant at all scales. However, AUTODISC is able to automatically identify a reasonably good teacher assistant in a much more efficient manner compared to MANDISC.

Sufficiency of One Teacher Assistant To examine whether one teacher assistant is sufficient, we insert more than one teacher assistant to AUTODISC and present the results in Figure 3(b). It is
clear that there is no obvious performance gain when applying more than one teacher assistant (two and three) in schedules. Therefore, we alternatively choose to use only one teacher assistant in AUTO DISC for training efficiency based on the sufficiency. The conclusion still holds for EncT5, which is supplied in Appendix J.

Impact of Candidate Sampling  We then study the impact of the sandwich-optimization in AUTO DISC by varying the number of sampled candidates $\eta$, and measuring the training cost and the student performance. From Table 4 we show the assembled sandwich together with sub-sampled fillings brings acceptable performance detriment and efficiency gain. In comparison with MAN DISC which we conduct 10 trials with different teacher assistants, AUTO DISC with just one candidate is able to achieve similar performance with much less training time.

Impact of $\lambda$  To show $\lambda$-Tradeoff is robust on the value of $\lambda$, we vary $\lambda$ within $\{0.1,0.2,0.3\}$. It can be seen from Table 5 that the performance of AUTO DISC is relatively stable with different values of $\lambda$. Moreover, we offer a $\lambda$-independent solution using a negative derivative of performance to scale as the tradeoff measure, which yields slightly worse results, as supplied in Appendix K.

4 Related Work

Language Model  Language models (LMs) [1, 5] are widely adopted in various natural language tasks [39, 40]. Typical LMs consist of a stack of transformer [41] encoder/decoder layers. Each encoder layer has two modules. The first is a self-attention module, and the second is a feed-forward module. A residual connection is employed around each of these modules, with a layer normalization placed either in (pre-norm) or out of (post-norm) the connection [42]. Each decoder layer additionally has a cross-attention module between the self-attention and feed-forward modules. While LMs exhibit excellent performance in various downstream tasks, their scales impede the deployment in
real-world applications. Therefore, it is an important research problem of learning compact language models from the large ones. In our work, we aim to make LMs deployable via model compression.

**Model Pruning** Inspired by the idea that not all parameters contribute equally to the overall performance of a model, model pruning [43] is widely adopted to waive the parameters with little impact. Model pruning spans from unstructured pruning [19, 44, 46] to structured pruning [11, 12, 18, 22, 47]. Unstructured pruning prunes parameters at neuron level referring to parameter magnitude [43, 44] or learning dynamics [45], while structured pruning [11, 22] prunes parameters at module level relying on parameter sensitivity to performance. Although unstructured pruning enjoys a finer-grained pruning, it can only fit specialized devices. In contrast, structured pruning generally fits modern acceleration devices. In our work, we adopt structured pruning for deriving the structures of candidates for its benefits for distillation. Pruning also offers an opportunity to optimize the efficiency of our method due to its merits [18, 19, 23, 24].

**Knowledge Distillation** Knowledge distillation [34] is employed as a promising choice for model compression. Knowledge distillation can be divided into two categories: task-specific [13, 34, 48, 35] and task-agnostic [14, 49, 51, 56, 25] distillation. Task-specific methods distill fine-tuned models with task-specific data, while task-agnostic methods distill pre-trained models directly with task-agnostic data. Distillation objective is central to both task-specific and task-agnostic distillation, and distilling logits [34] is the most common way. Recently, hidden states [50, 51], attention distributions [14, 48, 25], and high-order relations [35] are taken into consideration for better abstraction. To combine the power of both pre-training and fine-tuning, two-stage distillation [36] appends an enhancing distillation stage after the task-agnostic distillation. Teacher assistant-based distillation [14, 15, 17] is showcased to trade in teacher scale for student performance by inserting an intermediate-scale teacher assistant. This phenomenon is also supported in other work that better student performance should be attained with slightly worse teacher learning capacity [52]. However, setting the teacher assistant to a small scale with high performance for the student is nontrivial. A manual distillation schedule can produce an inferior scale-performance tradeoff. Although there is a related justification for an automatic distillation schedule [15], it is based on strong assumptions and has not yet been verified.

5 Conclusions

In this paper, we propose AUTODISC to automatically identify an effective teacher assistant for teacher assistant-based distillation, which bridges the large capacity gap between the teacher and student. Based on the largely-neglected observation that both the performance and the scale of the teacher assistant are of great importance to the performance of the student, we introduce a $\lambda$-Tradoff measure that balances the scale and the performance of the teacher assistant, and show that it is positively correlated with the final student performance. To compute the measures for possible teacher assistant candidates, we leverage gridding and pruning to specify these candidates and achieve an once-for-all optimization for these candidates based on two properties. The best teacher assistant is selected according to the $\lambda$-Tradoff value. Comprehensive experiments demonstrate the improved performance and applicability of AUTODISC. Experimental results on a language model over one billion parameters show the scalability of AUTODISC.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019. URL [https://doi.org/10.18653/v1/n19-1423](https://doi.org/10.18653/v1/n19-1423).

[2] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *arXiv*, abs/1907.11692, 2019. URL [http://arxiv.org/abs/1907.11692](http://arxiv.org/abs/1907.11692).

[3] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *Preprint*, 2019. URL [https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf](https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf).
[18] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *ICLR*, 2017. URL https://openreview.net/forum?id=rJqFGTs1g

[19] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *ICLR*, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7

[20] Frederick Liu, Siamak Shakeri, Hongkun Yu, and Jing Li. Enct5: Fine-tuning T5 encoder for non-autoregressive tasks. *CoRR*, abs/2110.08426, 2021. URL https://arxiv.org/abs/2110.08426

[21] Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *ICLR*, 2020. URL https://openreview.net/forum?id=Sy102yStDr

[22] Mengzhou Xia, Zexuan Zhong, and Danqi Chen. Structured pruning learns compact and accurate models. *arXiv*, abs/2204.00408, 2022. URL https://doi.org/10.48550/arXiv.2204.00408

[23] Jiahui Yu and Thomas S. Huang. Universally slimmable networks and improved training techniques. In *ICCV*, pages 1803–1811, 2019. URL https://doi.org/10.1109/ICCV.2019.00189

[24] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. In *ICLR*, 2020. URL https://openreview.net/forum?id=HylxE1HKwS

[25] Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *ACL-IJCNLP*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 2140–2151, 2021. URL https://doi.org/10.18653/v1/2021.findings-acl.188

[26] Richard Socher, Alex Perelygin, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642, 2013. URL https://aclanthology.org/D13-1170/

[27] William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *IWP@IJCNLP*, 2005. URL https://aclanthology.org/I05-5002/

[28] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*, pages 1112–1122, 2018. URL https://doi.org/10.18653/v1/n18-1101

[29] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392, 2016. URL https://doi.org/10.18653/v1/d16-1264

[30] Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. The fifth PASCAL recognizing textual entailment challenge. In *TAC*, 2009. URL https://tac.nist.gov/publications/2009/additional.papers/RTE5_overview.proceedings.pdf

[31] Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *KR*, 2012. URL http://www.aaai.org/ocs/index.php/KR/KR12/paper/view/4492
[34] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv*, abs/1503.02531, 2015. URL http://arxiv.org/abs/1503.02531

[35] Geondo Park, Gyeongman Kim, and Eunho Yang. Distilling linguistic context for language model compression. In *EMNLP*, pages 364–378, 2021. URL https://doi.org/10.18653/v1/2021.emnlp-main.30

[36] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. In *EMNLP*, volume EMNLP 2020 of *Findings of ACL*, pages 4163–4174, 2020. URL https://doi.org/10.18653/v1/2020.findings-emnlp.372

[37] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *NeurIPS*, pages 525–536, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/432aca3a1e3c45e339f35a30c8f65edce-Abstract.html

[38] Xi Lin, Hui-Ling Zhen, Zhenhua Li, Qingfu Zhang, and Sam Kwong. Pareto multi-task learning. In *NeurIPS*, pages 12037–12047, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/685b6fde03eb646c27ed565881917c71c-Abstract.html

[39] Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. PLATO: pre-trained dialogue generation model with discrete latent variable. In *ACL*, pages 85–96, 2020. URL https://doi.org/10.18653/v1/2020.acl-main.9

[40] Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. PEGASUS: pre-training with extracted gap-sentences for abstractive summarization. In *ICML*, volume 119 of *PMLR*, pages 11328–11339, 2020. URL http://proceedings.mlr.press/v119/zhang20ae.html

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee9fbd053c1c4a845aa-Abstract.html

[42] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture. In *ICML*, volume 119 of *PMLR*, pages 10524–10533, 2020. URL http://proceedings.mlr.press/v119/xiong20b.html

[43] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *NeurIPS*, pages 1135–1143, 2015. URL https://proceedings.neurips.cc/paper/2015/file/ae0eb33ed392bcefe64622b2499a056fe-Paper.pdf

[44] Christos Louizos, Max Welling, and Diederik P. Kingma. Learning sparse neural networks through 1_0 regularization. In *ICLR*, 2018. URL https://openreview.net/forum?id=H1Y8hng0b

[45] Victor Sanh, Thomas Wolf, and Alexander M. Rush. Movement pruning: Adaptive sparsity by fine-tuning. In *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/ea15aabaa768ae4a5993a844f4fa6e4-Abstract.html

[46] Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. The lottery ticket hypothesis for pre-trained BERT networks. In *NeurIPS*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/b6af2c9703f203a2794be903d443a2f3e3-Abstract.html

[47] François Lagunas, Ella Charlaix, Victor Sanh, and Alexander M. Rush. Block pruning for faster transformers. In *EMNLP*, pages 10619–10629, 2021. URL https://doi.org/10.18653/v1/2021.emnlp-main.829

[48] Jianquan Li, Xiaokang Liu, Honghong Zhao, Ruifeng Xu, Min Yang, and Yaohong Jin. BERT-EMD: many-to-many layer mapping for BERT compression with earth mover's distance. In *EMNLP*, pages 3009–3018, 2020. URL https://doi.org/10.18653/v1/2020.emnlp-main.242
A Technical Details of Pruning

Concretely, following previous work \[11\], the pruning always starts with the least important parameters/features, which are identified according to importance scores. The importance scores are approximated by first masking the parameterized structures. \( \mu^{(i)} \), \( \nu^{(i)} \), and \( \xi^{(j)} \) denote the mask variables respectively for a self-attention head, optionally a cross-attention head, and a feed-forward neuron, such that for an intermediate input \( X \) and potentially an encoder-produced input \( E \):

\[
Z^{o} = \text{SelfAttention}^o(X) = \sum_{i}^{h} \mu^{(i)} \text{softmax}(XW^{(i)}_QW^{(i)}_KX^\top)XW^{(i)}_VW^{(i)}_O,
\] (4)

\[
Z^{o} = \text{CrossAttention}^o(Z, E) = \sum_{i}^{h} \nu^{(i)} \text{softmax}(Z^{o}W^{(i)}_QW^{(i)}_K,E^\top)EW^{(i)}_VW^{(i)}_O,
\] (5)

\[
\tilde{X}^{o} = \text{FeedForward}^o(Z^{o}) = \sum_{j}^{d} \xi^{(j)} g(Z^{o}W^{(j)}_{1})W^{(j)}_{2},
\] (6)

where potential bias terms (e.g., linear bias and position bias) are omitted, \( i \) means \( i \)-th head among \( h \) heads, \( j \) means \( j \)-th intermediate neuron among \( d \) neurons, and \( g \) is an activation function. We initialize all mask variables to ones to preserve the original structure at the very beginning.

Then expected absolute gradients over either fine-tuning or pre-training data gives the important scores:

\[
\mathbb{E}_{(x,y)\sim D} \left| \frac{\partial \mathcal{L}(x,y)}{\partial \mu^{(i)}} \right|, \mathbb{E}_{(x,y)\sim D} \left| \frac{\partial \mathcal{L}(x,y)}{\partial \nu^{(i)}} \right|, \mathbb{E}_{(x,y)\sim D} \left| \frac{\partial \mathcal{L}(x,y)}{\partial \xi^{(j)}} \right|,
\] (7)

where \( (x, y) \) is a data point and \( \mathcal{L} \) is the task-specific loss for task-specific models or the language modeling loss for pre-trained models. \( \mathbb{E} \) represents expectation. The absolute value of gradient for a mask indicates how large the impact of pruning the corresponding structure is, thus implying how important the structure is.

Intuitively, we take a global ranking, in contrast to a local one as in other literature \[12\], for the structures of the same type (i.e., attention head or feed-forward element) from all stacking layers for pruning preference, before which we also normalize the importance scores for same-type structures in a layer with \( \ell_2 \) norm, as suggested by Molchanov et al. \[53\], for a balanced pruning. Therefore, for each candidate, we separately prune attention heads and feed-forward elements to the scale so that we reach a qualified structure. For the sake of a corner case that all structures in a module are pruned, we skip the module by feeding the input as the output. While we can alternate to an quite recent pruning method \[22\] exploiting both coarse-grained and fine-grained strategies for state-of-the-art performance, we argue that our framework is agnostic to pruning methods and keep the pruning method simple.
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        |
| Wikipedia | 35M          | -          | 128         | -               |

Table 7: The hyperparameters for both task-specific and task-agnostic distillation. The learning rate is searched within different grids for BERT\textsubscript{base} and EncT5\textsubscript{1.5b}.

| Hyperparameter                  | Task-specific Distillation | Task-agnostic Distillation |
|---------------------------------|---------------------------|----------------------------|
| Batch Size                      | [16, 32]                  | $8 \times 128 = 1024$     |
| Optimizer                       | AdamW                     | AdamW                      |
| Learning Rate                   | {1e-5, 2e-5, 3e-5}        | {1e-4, 2e-4, 3e-4}        |
| Training Epochs                 | 10                        | 5                          |
| Early-stop Epochs               | 5                         | -                          |
| Warmup Proportion               | 0.1                       | 0.01                       |
| Weight Decay                    | 0.01                      | 0.01                       |
| Sampling Number $\eta$          | 9                         | 1                          |

B Dataset Statistics

We conduct experiments on seven datasets. The detailed statistics, maximum sequence lengths, and metrics for datasets we use are shown in Table 6, where the Wikipedia corpus used for pretraining is also attached.

C Additional Implementation Details

The summary of hyperparameters for both task-specific and task-agnostic distillation is shown in Table 7. We will be releasing our code and scripts in the final version for exact reproducibility.

D Additional Results on BERT\textsubscript{base}

We further conduct experiments on extremely small scale student model, i.e., BERT$_{3\%}$. The results are shown in Table 8.

E Inference Measurement

Since FLOPs only offers theoretical inference compute, we additionally provide throughput for empirical inference compute of each model with throughput (i.e., processed tokens per micro second) in Table 9. The test environment is established by feeding $32 \times 128$ tokens to models. The amount of decomposed parameters is also attached for a reference.

F Pruned Structure Distributions

We give the distributions of example pruned structures in Figure 4, which exactly show what pruned LMs consist of. While pruned BERT$_{base}$ tends to preserve bottom and middle layers, pruned EncT5$_{1.5b}$ tends to preserve bottom layers. Meanwhile, neurons in feed-forward layers are more likely to be pruned than heads in attention layers, owing to the centrality of the attention module within an transformer layer.
Table 8: Additional results of task-specific distillation upon BERT\textsubscript{base}.

| Model | FLOPs | SST-2 | MRPC | STS-B | QQP | MNLI-m/mm | QNLI | RTE | Average |
|-------|-------|-------|------|-------|-----|-----------|------|-----|---------|
| BERT\textsubscript{ps} - \mathcal{L}_\text{TSD} | 0.3G  | 85.2  | 83.6 | 81.9  | 82.1| 71.9/72.7 | 81.9 | 57.4| 77.1    |
| w/ M\textsc{ANDISC} \cite{13} | 0.3G | 85.6  | 85.0 | 82.7  | 82.7| 72.7/72.8 | 82.0 | 59.6| 77.9    |
| w/ AUTO\textsc{DISC} | 0.3G | 85.9  | 85.7 | 83.6  | 83.1| 72.9/73.6 | 81.9 | 58.1| 78.1    |

Table 9: Inference compute measurement.

| Model | FLOPs | Throughput | Trm params | Emb params |
|-------|-------|------------|------------|------------|
| BERT\textsubscript{base} | 10.9G | 55.7tokens/ms | 85.7M | 23.8M |
| BERT\textsubscript{10%} | 1.1G | 278.2tokens/ms | 9.1M | 23.8M |
| BERT\textsubscript{5%} | 0.5G | 412.9tokens/ms | 4.9M | 23.8M |
| BERT\textsubscript{large} | 38.7G | 17.6tokens/ms | 303.3M | 31.8M |
| BERT\textsubscript{10%} | 3.9G | 104.1tokens/ms | 31.3M | 31.8M |
| BERT\textsubscript{5%} | 1.9G | 154.2tokens/ms | 16.3M | 31.8M |
| EncT5\textsubscript{1.5b} | 155.8G | 4.8tokens/ms | 1275.1M | 32.9M |
| EncT5\textsubscript{10%} | 15.6G | 38.8tokens/ms | 127.4M | 32.9M |
| EncT5\textsubscript{5%} | 7.8G | 64.0tokens/ms | 64.0M | 32.9M |

G Data Augmentation for TinyBERT

We compare TinyBERT with and without data augmentation as in Table 10. The results with data augmentation are retrieved from the original paper, since the augmented data is not publicly available. The results demonstrate that TinyBERT is largely supported with data augmentation for good performance.

Table 10: The results of TinyBERT with and without DA.

| Model | FLOPs | SST-2 | MRPC | STS-B | QQP | MNLI-m/mm | QNLI | RTE | Average |
|-------|-------|-------|------|-------|-----|-----------|------|-----|---------|
| TinyBERT\textsubscript{4L|312H} \cite{36} | 0.6G | 88.3 | 88.5 | 84.3 | 84.0 | 77.0/77.4 | 82.5 | 63.5 | 80.7    |
| w/ DA \cite{36} | 0.6G | 92.7 | 90.2 | 86.3 | 87.1 | 82.8/82.8 | 88.0 | 65.7 | 84.5    |
| MiniLMv2\textsubscript{3L|384H} \cite{25} | 0.7G | 89.1 | 89.1 | 86.6 | 85.4 | 77.8/78.4 | 87.2 | 66.1 | 82.5    |

H Results on BERT\textsubscript{large}

We show extended results of AUTO\textsc{DISC} on BERT\textsubscript{large} for readers’ interest in Table 11. Consistent patterns have been observed as in BERT\textsubscript{base}.

I Results on Small-scale LMs

When AUTO\textsc{DISC} is applied to small-scale MiniLM\textsubscript{12|384H} and BERT\textsubscript{mini} as shown in Table 12, AUTO\textsc{DISC} can reversely affect the performance of conventional distillation. Contrarily, MAN\textsc{DISC} can still improve or at least retain the performance. However, it is less necessary to compress small-scale LMs.

J Varying Schedules for EncT5

Performance variations among possible schedules for EncT5 are displayed in Figure 5, where the existence of scale-performance tradeoff and sufficiency of one teacher assistant can be verified.

K Negative Derivative-Tradeoff

As mentioned in the main paper, although $\lambda$-Tradeoff is able to provide stable tradeoff measurement, it is dependent on the value of $\lambda$. To eliminate this dependency, we design a new measure, negative
Figure 4: The distributions of example pruned structures. The structures are derived with MRPC dataset.

dervative-Tradeoff, which computes the negative derivative of performance to scale at each candidate scale as: \( t_a = \lim_{\delta \to 0} -\frac{m_{a+\delta} - m_a}{\Delta m_a}. \) In the discrete case, \( t_{a_i} = -\frac{(m_{a_i+1} - m_{a_i})}{\Delta m_{a_i}}. \) The idea of the measure is basically derived from saving the performance from a potentially significant drop. However, first-order estimation can lead to a high estimation variance and can be further tuned with second-order or so for better performance. The comparison results using \( \lambda \)-Tradeoff and ND-Tradeoff are shown in Table 13. It can be seen from the table that AUTODISC-ND also achieves comparable results.
Table 11: The results of task-specific distillation upon BERT\textsubscript{large}.

| Model           | FLOPs | SST-2 | MRPC | STS-B | RTE  | Average |
|-----------------|-------|-------|------|-------|------|---------|
| BERT\textsubscript{base} | 10.9G | 93.8  | 91.5 | 87.1  | 71.5 | 86.0    |
| BERT\textsubscript{10\%-L\textsubscript{TSD}} w/ MANDisc [15] | 1.1G  | 88.8  | 87.8 | 84.0  | 66.4 | 81.8    |
| BERT\textsubscript{10\%-L\textsubscript{TSD}} w/ AUTODisc | 1.1G  | 89.0  | 88.2 | 84.8  | 66.8 | 82.2    |
| BERT\textsubscript{5\%-L\textsubscript{TSD}} w/ MANDisc | 0.5G  | 85.4  | 85.5 | 83.9  | 63.2 | 79.5    |
| BERT\textsubscript{5\%-L\textsubscript{TSD}} w/ AUTODisc | 0.5G  | 86.1  | 87.0 | 84.1  | 65.7 | 80.7    |
| BERT\textsubscript{large} | 38.7G | 94.2  | 92.5 | 90.1  | 75.5 | 88.1    |
| BERT\textsubscript{10\%-L\textsubscript{TSD}} w/ MANDisc | 3.9G  | 90.4  | 88.1 | 87.0  | 66.1 | 82.9    |
| BERT\textsubscript{10\%-L\textsubscript{TSD}} w/ AUTODisc | 3.9G  | 90.5  | 88.8 | 87.8  | 66.1 | 83.3    |
| BERT\textsubscript{5\%-L\textsubscript{TSD}} w/ MANDisc | 1.9G  | 89.2  | 85.7 | 85.8  | 61.4 | 81.2    |
| BERT\textsubscript{5\%-L\textsubscript{TSD}} w/ AUTODisc | 1.9G  | 89.6  | 87.4 | 87.3  | 61.4 | 81.4    |

Table 12: The results of task-specific distillation upon small-scale LMs.

| Model          | FLOPs | SST-2 | MRPC | STS-B | QQP | MNLI-m/mm | QNLI | RTE | Average |
|----------------|-------|-------|------|-------|-----|-----------|------|------|---------|
| MiniLM\textsubscript{12L/384H} | 2.72G | 92.1  | 90.9 | 88.6  | 87.2| 83.0/83.3| 90.7 | 72.9 | 86.1    |
| MiniLM\textsubscript{10\%-L\textsubscript{TSD}} w/ MANDisc [15] | 0.26G | 87.8  | 87.1 | 85.6  | 84.3| 77.2/78.4| 84.8 | 66.4 | 81.5    |
| MiniLM\textsubscript{10\%-L\textsubscript{TSD}} w/ AUTODisc | 0.26G | 88.2  | 88.2 | 86.3  | 84.7| 77.8/79.2| 85.2 | 65.7 | 81.9    |
| BERT\textsubscript{mini} | 0.60G | 87.5  | 86.4 | 85.3  | 85.0| 76.1/77.2| 84.5 | 66.8 | 81.1    |
| BERT\textsubscript{10\%-L\textsubscript{TSD}} w/ MANDisc | 0.04G | 83.3  | 83.8 | 81.6  | 81.6| 66.3/71.4| 82.7 | 58.8 | 76.2    |
| BERT\textsubscript{10\%-L\textsubscript{TSD}} w/ AUTODisc | 0.04G | 83.8  | 84.1 | 80.7  | 82.0| 66.4/71.6| 82.9 | 58.1 | 76.2    |

Figure 5: Performance comparisons among various schedules for EncT5. The dotted lines represent performance variations using either one or two teacher assistants for MANDisc. The triangles represent performance resulting from AUTODisc using one teacher assistant.
Table 13: The results of negative derivative-Tradeoff upon BERT<sub>base</sub>.

| Model               | FLOPs | SST-2 | MRPC | STS-B | RTE  | Average |
|---------------------|-------|-------|------|-------|------|---------|
| BERT<sub>base</sub> | 10.9G | 93.8  | 91.5 | 87.1  | 71.5 | 86.0    |
| BERT<sub>10%</sub>-L<sub>TSD</sub> | 1.1G  | 88.8  | 87.8 | 84.0  | 66.4 | 81.8    |
| w/ MANDisc [15]     | 1.1G  | 89.0  | 88.2 | 84.8  | 66.8 | 82.2    |
| w/ AutoDISC-λ       | 1.1G  | 89.1  | 88.4 | 85.4  | 68.2 | 82.7    |
| w/ AutoDISC-ND      | 1.1G  | 89.8  | 87.9 | 85.4  | 66.4 | 82.4    |
| BERT<sub>5%</sub>-L<sub>TSD</sub> | 0.5G  | 85.4  | 85.5 | 83.9  | 63.2 | 79.5    |
| w/ MANDisc [15]     | 0.5G  | 86.1  | 87.0 | 84.1  | 65.7 | 80.7    |
| w/ AutoDISC-λ       | 0.5G  | 86.9  | 87.6 | 84.8  | 66.8 | 81.5    |
| w/ AutoDISC-ND      | 0.5G  | 86.8  | 86.0 | 84.9  | 66.8 | 81.1    |