# ESEAD: AN ENHANCED SIMPLE ENSEMBLE AND DISTILLATION FRAMEWORK FOR NATURAL LANGUAGE PROCESSING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large-scale pre-trained language models (PLM) are today's leading technology for a wide range of natural language processing tasks. However, the enormous size of these models may discourage their use in practice. To tackle this problem, some recent studies have used knowledge distillation (KD) to compress these large models into shallow ones. Despite the success of the knowledge distillation, it remains unclear how students learn. We extend knowledge distillation in this paper and propose an enhanced version of the logits-based distillation method, ESEAD, to utilize the knowledge of multiple teachers to assist student learning. In extensive experiments with total 13 tasks on the GLUE and SuperGLUE benchmarks, ESEAD with different fine-tuning paradigms (e.g., delta tuning) obtained superior results over other KD methods and even outperformed the teacher model on some tasks. In addition, ESEAD remained the best performing student model in the few-shot (e.g., 100 samples) settings.

## 1 INTRODUCTION

PLMs have been extensively studied through the design of new pretexts, architectures, or attention mechanisms. On a variety of natural language processing tasks, these PLMs perform far better than traditional methods. Nevertheless, their drawbacks are still evident, such as reasoning on a limited number of devices and limiting the applicability of these models in real-world scenarios.

To alleviate the aforementioned limitations, KD is proposed to transfer the knowledge via minimizing KL-Divergence between prediction logits between larger teacher models and smaller student models. Recently, most research attention has been drawn to distill knowledge from two kinds: pre-training and distilling a task-agnostic model that can be used to fine-tune arbitrary natural language downstream tasks; and a task-specific model that progressively accumulates intermediate layers of knowledge. Even though the task-agnostic approach has the flexibility of fine-tuning arbitrary downstream tasks, it requires at least the same amount of data and sufficient computational power as the pre-trained teacher model. When a model is task-specific distilled, it can achieve significantly higher compression and faster inference rates at the cost of increased training time due to hierarchical learning, e.g., learning weights by gradually unfreezing layers in the Transformer model.

From an intuitive standpoint, logits distillation based methods should achieve comparable performance as progressive knowledge distillation due to the nature of logits that they are trained to represent a high level of semantic content. For some reason, however, its potential remains inactive. In this study, we investigate logits-based knowledge distillation and propose our framework ESEAD which leverages the mixed logits from multiple teachers to further improve student model's generalization. Overall, the summary of contributions in this paper is listed as following:

- We propose a new distillation strategy to encourage the transfer of linguistic knowledge encoded in teacher models to students.
- We apply our techniques on extensive experiments on total 13 tasks among GLUE and SuperGlue benchmarks, and achieved superior results comparing to other KD methods.
- We investigate our method in few-shot settings, and it consistently perform the best.

- Our framework is simple, architecture agnostic, and can be applied to different types of fine-tuning methods e.g. delta tuning (Ding et al., 2022a) and self-distillation settings.

Finally, we will publish the code and release model weights for the tasks in GLUE and SuperGLUE.

## 2 KNOWLEDGE DISTILLATION

### 2.1 RELATED WORK

Hinton et al. (2015) first proposed a prototype for KD that transfers knowledge from a large teacher network $T$ to a small student network $S$. Student model $S$ receives two guidance signals: first, the training data set, also known as hard labels $y$, and secondly, the teacher network predictions $f^T$, known as soft targets. Mathematically, the loss objective $\mathcal{L}_{total}$ of the student network is a weighted sum of the task loss $\mathcal{L}_{task}$, which minimizes the discrepancy between the target output and the student output $f^S$, and the knowledge distillation loss $\mathcal{L}_{KD}$, which minimizes the distance between the teacher and the student output.

$$\begin{aligned} \mathcal{L}_{total}(x, y) &= \alpha \mathcal{L}_{task} + \lambda \mathcal{L}_{KD} \\ \mathcal{L}_{task} &= \text{CE}(f^S(x), y) \\ \mathcal{L}_{KD} &= \mathcal{T}^2 \text{KL}(f^S(x), f^T(x); \mathcal{T}) \end{aligned} \tag{1}$$

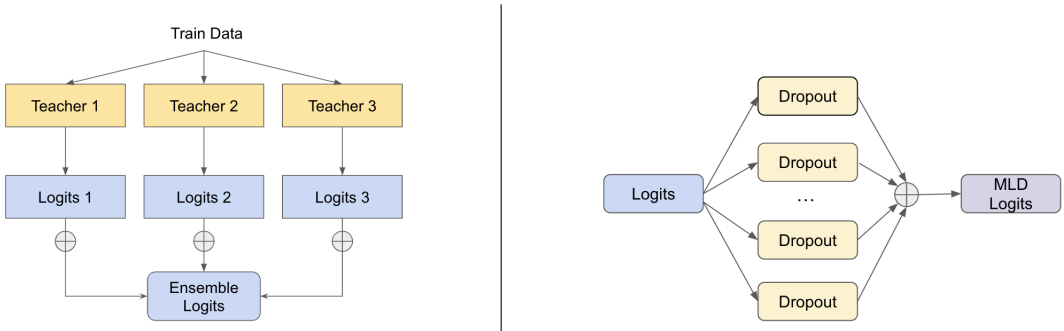
In the initial KD, $\alpha$ is computed as a $1 - \lambda$, while $\tau$ is the hyper-parameter named temperature, which is used to smooth the distribution of the output. Guo et al. (2020) replaces the logits of teachers by normalizing and multiplying by the average $l_2$-norm for distilling a student model. Jafari et al. (2021) leverages annealing function of temperature to adjust the steps of student learning.

Most of research work adopt progressive distilling by converting Equation 1 from prediction to multi-layer feature distillation shown as Equation 2, where there are $M$ layers feature to utilize, and $\lambda_m$ is the hyper-parameter that represents the importance of $m$-th layer's distillation.

$$\mathcal{L}_{KD} = \sum_{m=0}^{M+1} \mathcal{L}_m(f_m^S(x), f_{g(m)}^T(x); \lambda_m) \tag{2}$$

Jiao et al. (2020) and Sun et al. (2019) utilize intermediate features through progressive learning, thawing the Transformers layer sequentially from top to bottom, and sharing the output of each Transformer between the teacher and students for distillation. The same approach was adopted in the work (Jiao et al., 2020), while the only difference was the invocation of a two-stage distillation where it firstly distilled a task-agnostic student model, and then it can be applied for language related downstream tasks. Unlike leveraging intermediate features as guidance, Wang et al. (2020) considers last layer strategy. Mukherjee et al. (2021) attempts to minimize the mean squared errors between the attention patterns and hidden states of the last layer between the student and the teacher. The results of work (Wang et al., 2020) suggest that a deep self-attention distillation which minimizes KL divergence over the last layer's attention transfer and value-relation transfer, thereby deeply mimicking the self-attention behavior of the top Transformer layer. Chen et al. (2021) relies on the proposed importance metric to extract the teacher's parameters in a randomized manner to obtain a student model of arbitrary length and width.

The mainstream approach uses a single teacher distillation setup, while Wu et al. (2021) and Liu et al. (2021)show that incorporating multiple teacher models into knowledge distillation has the potential to learn better student models. Most of the approaches in this line use PLMs with different architectures to provide logits for students to improve their performance. However, logits based on PLMs with different architectures are abstract. In other words, the student model is not able to learn directly from the hybrid knowledge. To overcome this, some additional designs are used to ease the uptake by students. For example, Wu et al. (2021) proposes a multi-teacher common network tuning framework with a shared pool and prediction module to jointly fine-tune multiple teacher models to adjust their output hidden states for better co-teaching of students. Unlike the use of additional designs to improve logits-based distillation, we followed the work of Mei & Sroch (2022), which utilizes the ensemble knowledge of multiple teachers and learning strategies to allow the student model to improve its generalization.




(a) Multi-Teacher Ensemble

(b) Multi-Logits Dropout

Figure 1: Ensemble Illustration

### 2.2 METHOD

In this section, we present the details of our work ESEAD: **E**nhanced **S**imple **E**nsemble and Knowledge **D**istillation. ESEAD is an enhanced version of the work (Mei & Sroch, 2022). First, we will review some of the mechanisms that have been carried over, and then introduce the newly proposed parts on them.

Similarly, ESEAD employs a multi-teacher knowledge distillation strategy, as shown in Figure 1a. Specifically, there are two methods to provide the mixed knowledge for student to learn, which are weighted and random. Their formula representations are show in Equation 3 & 4 respectively. The weighted based method produce a weighted average over logits of multiple teachers where their weights $w_i$ are drawn from a pre-defined Dirichlet distribution for every batch. In this setup, the student model is allowed to distribute attention to different teachers' instructions at each training step. On the other hand, the random method can be seen as a hard version of the weighted method because it allows the student model to learn from the knowledge of only one of the teachers instead at every training step. The $\mathbf{t}$ in Equation 4 refers to the chosen teacher in the single piece of training data.

$$f^T_{weighted}(x) = \frac{1}{T}\sum_{i=1}^{T} w_i f_i^T(x;\theta_i), \tag{3}$$

$$f^T_{random}(x) = \sum_{i=1}^{T} \mathbb{1}(\mathbf{T}=\mathbf{t}) f_i^T(x;\theta_i) \tag{4}$$

Student performance can be further improved by scheduling when to provide aggregate knowledge from multiple teachers. Empirical studies have shown that adding knowledge to students after a threshold produces a double decent curve in terms of validation loss, resulting in smaller local minima. However, it requires a priori knowledge or hyper-parameter search to find the threshold for adding those ensemble knowledge to avoid divergence, which leads to additional costs. We investigated this further and suggested that there are two techniques that can enhance the generalization of the student model more at a lower cost, which are overlooking and multi-logits dropout.

**Overlooking**. Intuitively, the reason why the above approach is effective is that the student model does not exactly mimic the teacher's behavior in the distillation process, as the initial KD did, but selectively focuses on specific examples of the teacher's varying importance. Moreover, teacher models do not always produce the correct results, so students who receive both signals: the gold data distribution and the incorrect teacher output can cause disorder and thus produce incorrect results. Initial KD relies on a hyper-parameter $\lambda$ to control the ratio of the two signals to learn shown in Equation 1. However, this results in the knowledge of teachers who have a high level of confidence in the samples not being sufficiently accepted by the student. In this regard, we propose our new mechanism to improve students' proficiency. Specifically, we considered two choices of overlooking: informative and random.

Informational overlooking (I.O) is carried out intuitively by transforming the teacher's logits to scores via the Softmax transformation, and those scores are compared with a threshold $t$ to decide whether

their logits should be overlooked. $t$ is a hyper-parameter ranging from 0 to 1. When $t \to 1$, the student model learns more confident knowledge from teachers, and when $t \to 0$, the student model learns from the data distribution only where no distillation occurs.

On the other hand, we propose random overlooking (R.O) as a counter-intuitive method. During each epoch, we randomly select $\rho$ percent of the total batch for the student network to receive guidance only from the data distribution. The overlook rate $\rho$ controls the amount of guidance teachers provide to students. When $\rho \to 0$, the total loss converges to the standard total loss as in Equation 1, while when $\rho \to 1$, the total loss falls to the task loss. We found that $\rho$ is also very helpful when the gap between the capacity between the student and teachers.

**Multi-logits Dropout (MLD)** Another direction worth investigating is how to provide better ensemble logits through teachers. Inspired by Inoue (2019), we propose multi-logits dropout to implicitly increase the number of teachers to provide better aggregate knowledge shown in Figure 1b. In Equation 3 & 4, we replace each teacher's logits $f_i^T$ with their augmented version i.e., we apply $d$ dropout mask with a dropout proportion $r$ to each teacher logits, and then calculate its mean.

Overall, Algorithm 1 presents the steps for ESEAD distillation with random overlooking. To efficiently provide logits of teachers, we adapt parameter efficient learning paradigm such as delta tuning. More details of delta tuning are presented in section 3.2.

**Algorithm 1:** ESEAD Distillation

**Input:** trainData: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
initialize $\rho$, $d$, $r$
**for** *each epoch* **do**
  random sample overlook batches given $\rho$
  **for** *each batch b* **do**
    **if** *batch b in overlook batches* **then**
      train student with $D_b$
    **else**
      generate $f^T$ from multi-teachers
      apply MLD with $d$ and $r$ to get $f_{mld}^T$
      train student with $D_b$ and $f_{mld}^T$
    **end**
  **end**
**end**

## 3 EXPERIMENTS

In this section, we describe the experimental evaluation of our proposed ESEAD method. We evaluate our technique on 13 different natural language understanding tasks. Details on the datasets and experimental results are provided in the following sub-sections.

### 3.1 DATASETS

Our evaluation consisted of 8 tasks over the General Language Understanding Evaluation (GLUE) (Wang et al., 2019b) benchmark and 5 tasks over the SuperGLUE (Wang et al., 2019a) benchmark. In summary, GLUE consist of 2 single sentence tasks: COLA (Warstadt et al., 2019), SST-2 (Socher et al., 2013), 3 sentence similarity task: MRPC (Dolan & Brockett, 2005), STS-B (Cer et al., 2017), QQP (Iyer et al., 2017), and 3 natural language inference tasks: MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), and RTE (Bentivogli et al., 2009), while similarly, BoolQ (Clark et al., 2019), WSC (Levesque et al., 2012), WIC (Pilehvar & os'e Camacho-Collados, 2018), CB (De Marneff et al., 2019) and COPA (Roemmele et al., 2011) are tested in SuperGLUE.

### 3.2 EXPERIMENT SETUP

Two distilled students $\text{ESEAD}_{\texttt{L6-H384-A12}}$ and $\text{ESEAD}_{\texttt{L6-H256-A8}}$ are obtined under the guidance of the uncased base version of BERT. The teacher model is a 12-layer Transformer with 768 hidden

sizes and 12 attention heads, and the students are a 6-layer Transformer with 384 hidden sizes and 12 attention heads and 256 hidden sizes and 8 attention heads, respectively. Both students are initialized by XtremeDistillTransformers (XDT) (Mukherjee et al., 2021). To increase the efficiency of utilizing teachers, we consider delta tuning, which only finetunes a small portion of the model parameters while keeping the rest untouched, largely reducing both the computation and storage costs. Three delta-tuning methods are implemented which are Adapter (Houlsby et al., 2019), LoRa (Hu et al., 2021) and BitFit (Zaken et al., 2021). Adapter tuning injects lightweight modules between the layers, resulting in only a small number of additional task-specific trainable parameters; LoRa applies trainable rank decomposition matrices into each layer of the Transformer architecture; BitFit only allows biases to be fine-tuned. Comparisons among all are present in Table 1.

Table 1: Trainable Parameters Comparison

| | Full | Adapter | LoRa | BitFit |
|---|---|---|---|---|
| BERT | 110M | 1.54M | 0.29M | 0.11M |
| RoBERTa | 125M | 1.58M | 0.34M | 0.15M |

Overall, three teachers will be trained under different seeds for each task. During training, we randomly sample checkpoints after 50% of the total epochs instead of taking the best checkpoint to make better use of the ensemble. According to Zhang et al. (2021), we applied the revised version of the Adam (Kingma & Ba, 2015) optimizer to all tasks. Furthermore, we leverage early stopping to avoid over-fitting. More details about hyperparameters are added to the table 7 in the appendix, all of which are done via Bayesian search with RayTune [1] (Liaw et al., 2018). For the delta tuning scheme, only the delta parameters will be trained. Note that full fine tuning is implemented with Huggingface Transformer [2] (Wolf et al., 2020), adapter tuning is accomplished via AdapterHub [3] (Pfeiffer et al., 2020), and LoRa and BitFit utilize Open Delta[4] (Ding et al., 2022b).

### 3.3 Performance Evaluation

We compare the performance of our method with multiple sets of baselines. The first group is the 12-layer version of the teacher model BERT (Devlin et al., 2019) under different tuning schemes. The second group are feature-based methods e.g., DistilBERT (Sanh et al., 2020), TinyBERT (Jiao et al., 2020), MiniLM (Wang et al., 2020), and XDT (Mukherjee et al., 2021). Most of the feature-based methods are 6 layers Transformers, while MiniLM is a narrow 12 layers models.

It is worth mentioning we do not use the standard technique for fine-tuning RTE, MRPC, and STS-B, i.e., first fine-tuning the student model on MNLI and then fine-tuning the other downstream tasks according to its weights. For the delta tuning scheme, only the delta parameters will be trained. Results are summarized in Table 2. For ease of notation, $\text{BERT}^{i}_{\texttt{base}}$ refers to its model based on $i$-th kinds of fine-tuning, where $i \in \{\text{F}, \text{A}, \text{L}, \text{B}\}$ stand for Full, Adapter, LoRa, and BitFit tuning respectively. Likewise, our method is defined in the same way. Overall, our findings are 1) Our larger version model, $\text{ESEAD}^{\text{F}}_{\texttt{L6-H384-A12}}$, under full fine-tuning, outperforms all the GLUE and SuperGLUE tasks except for the Wic compared to other KD methods. On average, it is about 2% higher than the best of the feature based distillation methods. 2) $\text{ESEAD}_{\texttt{L6-H384-A12}}$ distilled with different tuning paradigms achieves comparable results to the BERT teacher model, and in 10 out of 13 cases the model achieves even better performance e.g., MNLI, MRPC, QQP, SST-2, STS, WNLI, BoolQ, CB, COPA, and WSC. The smaller model $\text{ESEAD}_{\texttt{L6-H256-A8}}$ is also able to achieve better performance than teachers over 7 tasks. 3) Across tuning paradigms, both of our models improve from initialization. In general, the fully tuned teacher produces the best results, Adapter and Lora produce similar results afterwards, while the BitFit-based one is usually the worst, which is consistent with the results of their paper. Among some tasks such as QQP, QNLI, MNLI, efficient tuning methods can also achieve comparable performance. 4) ESEAD prefers to improve the performance of small data sets. The possible reason for this is that MLD behaves similarly to augmentation techniques and therefore it leads to higher generalization. To valid this point, we conduct experiments over few-shot

[1] https://github.com/ray-project/ray

[2] https://github.com/huggingface/transformers

[3] https://adapterhub.ml/

[4] https://github.com/thunlp/OpenDelta

Table 2: GLUE and SuperGLUE Benchmarks Comparison on dev set. † Refers to the scores obtained from Hugginface/Open Delta implementation.

| | MNLI (m-Acc/mm-Acc) | MRPC (Acc/F1) | QNLI (Acc) | QQP (Acc) | RTE (Acc) | SST-2 (Acc) | STS (Pear/Spear) | WNLI (Acc) | BoolQ (Acc) | CB (Acc/F1) | COPA (Acc) | Wic (Acc) | WSC (Acc) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\text{BERT}^{\text{F}}_{\texttt{base}}$ | 84.5/- | -/87.3 | 91.7 | 91.3 | 68.6 | 93.2 | -/89.0 | 53.5 | 74.3† | 91.1/90.5† | 62.2† | 72.3† | 63.5† |
| $\text{BERT}^{\text{A}}_{\texttt{base}}$ | 84.4/84.5 | 84.6/89.3 | 91.4 | 90.5 | 71.8 | 92.0 | 88.8/88.4 | 53.5 | 75.5† | 83.9/77.5† | 57.0† | 62.2† | 61.6† |
| $\text{BERT}^{\text{L}}_{\texttt{base}}$ | 83.2/83.5 | 86.8/90.7 | 91.1 | 90.0 | 74.0 | 93.0 | 88.9/88.7 | 57.7 | 71.6† | 82.1/73.5† | 56.0† | 69.3† | 62.5† |
| $\text{BERT}^{\text{B}}_{\texttt{base}}$ | 82.2/82.9 | 84.6/89.4 | 90.2 | 85.2 | 71.5 | 92.3 | 88.4/88.1 | 57.7 | 71.2† | 82.1/73.5† | 55.0† | 67.1† | 61.5† |
| DistilBERT | 82.2/- | -/87.5 | 89.2 | 88.5 | 59.9 | 91.3 | -/86.9 | 56.3 | 73.5† | 91.1/93.4† | 54.0† | 65.2† | 63.5† |
| TinyBERT | 83.5/- | -/88.4 | 90.5 | 90.6 | 72.2 | 91.6 | -/89.6 | 56.3† | 77.8† | 83.9/75.4† | 63.0† | 68.8† | 62.5† |
| MiniLM | 84.0/- | -/88.4 | 91.0 | 91.0 | 71.5 | 92.0 | 88.8/88.7† | 56.3† | 73.9 | 76.8/53.6† | 54.0† | 69.6† | 64.4† |
| $\text{XDT}_{\texttt{L6-H256-A8}}$ | 82.4/83.0† | -/90.0 | 89.5 | 90.6 | 78.7† | 91.2 | 88.6/88.8† | 56.3† | 75.4† | 85.7/69.1† | 60.0† | 65.4† | 63.5† |
| $\text{XDT}_{\texttt{L6-H384-A12}}$ | 84.4/84.3† | -/90.0 | 90.3 | 91.0 | 80.9 | 92.3 | 89.9/90.0† | 56.3† | 76.8† | 87.5/81.8† | 64.0† | 65.8† | 63.5† |
| $\text{ESEAD}^{\text{F}}_{\texttt{L6-H256-A8}}$ | | | | | | | | | | | | | |
| + weighted | 83.1/83.4 | 89.7/92.6 | 90.0 | 91.0 | 79.4 | 92.0 | 89.6/89.8 | 59.2 | 77.1 | 92.9/89.4 | 63.2 | 67.4 | 64.4 |
| + random | 82.9/83.2 | 89.2/92.2 | 90.0 | 91.0 | 79.2 | 91.6 | 89.4/89.2 | 59.2 | 77.1 | 92.9/89.4 | 62.8 | 67.0 | 64.4 |
| $\text{ESEAD}^{\text{A}}_{\texttt{L6-H256-A8}}$ | | | | | | | | | | | | | |
| + weighted | 82.8/83.2 | 89.5/92.4 | 89.9 | 90.7 | 79.1 | 91.7 | 89.4/89.5 | 57.7 | 77.4 | 89.3/78.2 | 63.2 | 66.1 | 64.4 |
| + random | 82.9/83.2 | 88.0/91.4 | 89.7 | 90.6 | 79.1 | 91.3 | 89.4/89.3 | 57.7 | 77.1 | 87.5/78.7 | 62.8 | 67.4 | 64.4 |
| $\text{ESEAD}^{\text{L}}_{\texttt{L6-H256-A8}}$ | | | | | | | | | | | | | |
| + weighted | 82.6/83.2 | 89.0/92.3 | 89.7 | 90.7 | 79.1 | 91.7 | 89.3/89.2 | 56.3 | 76.8 | 85.7/69.1 | 62.5 | 66.3 | 64.4 |
| + random | 82.9/83.4 | 88.5/91.7 | 89.7 | 90.6 | 79.0 | 91.2 | 89.6/89.5 | 56.3 | 76.6 | 85.7/69.1 | 62.0 | 67.1 | 64.4 |
| $\text{ESEAD}^{\text{B}}_{\texttt{L6-H256-A8}}$ | | | | | | | | | | | | | |
| + weighted | 82.5/83.1 | 88.7/92.0 | 89.6 | 90.6 | 79.0 | 91.4 | 89.2/89.2 | 56.3 | 76.2 | 85.7/69.1 | 61.4 | 66.1 | 64.4 |
| + random | 82.5/83.1 | 88.0/91.3 | 89.6 | 90.5 | 78.7 | 91.2 | 89.5/89.4 | 56.3 | 76.2 | 85.7/69.1 | 61.4 | 66.9 | 64.4 |
| $\text{ESEAD}^{\text{F}}_{\texttt{L6-H384-A12}}$ | | | | | | | | | | | | | |
| + weighted | 85.4/85.6 | 91.4/93.9 | 91.2 | 91.3 | 82.4 | 92.8 | 90.5/90.4 | 59.7 | 78.3 | 94.8/93.1 | 68.4 | 68.0 | 65.8 |
| + random | 85.2/85.4 | 91.0/93.6 | 91.0 | 91.2 | 82.4 | 92.5 | 90.5/90.3 | 59.7 | 78.2 | 94.8/93.1 | 68.4 | 68.0 | 65.8 |
| $\text{ESEAD}^{\text{A}}_{\texttt{L6-H384-A12}}$ | | | | | | | | | | | | | |
| + weighted | 85.1/85.3 | 89.7/92.4 | 90.6 | 91.1 | 82.3 | 92.7 | 90.4/90.4 | 58.3 | 77.8 | 89.3/83.1 | 67.2 | 67.2 | 65.1 |
| + random | 85.0/85.2 | 88.7/91.8 | 90.7 | 91.1 | 81.6 | 92.5 | 90.4/90.3 | 58.2 | 77.4 | 89.3/83.1 | 66.0 | 67.2 | 64.9 |
| $\text{ESEAD}^{\text{L}}_{\texttt{L6-H384-A12}}$ | | | | | | | | | | | | | |
| + weighted | 85.0/85.1 | 89.5/92.5 | 90.7 | 91.1 | 82.3 | 92.7 | 90.4/90.3 | 58.3 | 77.5 | 87.5/81.8 | 67.0 | 67.2 | 65.1 |
| + random | 85.4/85.0 | 88.2/91.5 | 90.7 | 91.1 | 82.0 | 92.5 | 90.3/90.3 | 58.0 | 77.4 | 87.5/81.8 | 66.0 | 67.9 | 64.6 |
| $\text{ESEAD}^{\text{B}}_{\texttt{L6-H384-A12}}$ | | | | | | | | | | | | | |
| + weighted | 84.7/85.0 | 89.2/92.3 | 90.5 | 91.0 | 81.6 | 92.4 | 90.3/90.3 | 57.8 | 77.2 | 87.5/81.8 | 65.3 | 67.2 | 64.4 |
| + random | 84.9/85.1 | 88.7/92.0 | 90.5 | 91.0 | 81.1 | 92.3 | 90.3/90.2 | 57.7 | 77.1 | 87.5/81.8 | 65.0 | 67.1 | 64.3 |

settings, and results are show in Table 4. 5) For cases where the teacher model is not as good as the direct fine-tuning, SEAD can still improve.

**More Comparisons** ESEAD can be applied to any teacher structure. To illustrate, we distill student models initialized with the 6 layer DistilRoBERTa from two versions of RoBERTa (Liu et al., 2019). $\text{RoBERTa}_{\texttt{base}}$ is a 12 layer Transformer with 768 hidden states and 12 attention heads, while $\text{RoBERTa}_{\texttt{large}}$ is a 24 layer Transformer with 1024 hidden states and 16 attention heads. Following the same notation, $\text{ESEAD}^{\text{A}}_{\texttt{base}}$ refers to the student distilled from the $\text{RoBERTa}_{\texttt{base}}$, while $\text{ESEAD}^{\text{A}}_{\texttt{large}}$ is distilled from $\text{RoBERTa}_{\texttt{large}}$. For the sake of simplicity, we consider adapter tuning for teachers' efficient inference. Two extra baselines are included for comparison, which are DistilRoBERTa and AKD. Similar to them, we consider the standard technique of using MNLI-trained students to further distill the student model on the RTE task. As shown in Table 3, both of the ESEAD can further improve the performance of DistilRoBERTa, and they can achieve comparable performance with the teacher RoBERTa under adapter tuning in tasks such as MRPC and CB. With appropriate hyper-parameters, both methods outperform the AKD method except for the random based multi-teacher ensemble. The random based multi-teacher ensemble allows the student model to learn from only one of the teachers in each training step, so it may lead to a lack of feedback from the teacher, resulting in poorer generalizations. The results also suggest that the large gap between teacher and student competencies can be bridged by ESEAD and lead to additional performance payoffs, which also mitigate the use of teaching assistants (Mirzadeh et al., 2020).

**Few-shot Setting**. Real-world applications are often limited by the availability of data. Thus, we would like to test our method in few-shot settings. For simplicity, we chose six tasks for illustration and only show the case where the teacher is fully fine-tuned. For more experiments in few-shot settings such as delta-tuning paradigms, see Table 6. In each task, we randomly selected 100 samples from the original training set as the few-shot training data, and all models were validated using the associated full validation set, as well as their performance on the test set. We used the same hyper-parameters in Table 2, and results are obtained in Table 4. Unlike the full data scenarios, TinyBERT is the best among baselines on few-shot settings, and they achieved better scores than the teacher

Table 3: ESEAD Performance with RoBERTa as teachers.

| | MRPC (F1) | SST-2 (Acc) | RTE (Acc) | BoolQ (Acc) | CB (F1) | COPA (Acc) |
|---|---|---|---|---|---|---|
| $\text{RoBERTa}^{\text{A}}_{\text{base}}$ | 92.2 | 94.2 | 76.5 | 80.9 | 89.2 | 58.0 |
| $\text{RoBERTa}^{\text{A}}_{\text{large}}$ | 92.3 | 96.6 | 85.6 | 83.3 | 89.8 | 59.0 |
| DistilRoBERTa | 89.9† | 92.0† | 67.9† | 75.1† | 86.4† | 51.0† |
| $\text{DistilRoBERTa}_{\text{AKD}}$ | 90.6 | 93.1 | 73.6 | - | - | - |
| $\text{ESEAD}^{\text{A}}_{\text{base}}$ | | | | | | |
| + weighted | 92.8 | 93.2 | 73.8 | 77.6 | 90.5 | 56.5 |
| + random | 92.2 | 93.0 | 73.1 | 76.8 | 90.4 | 55.8 |
| $\text{ESEAD}^{\text{A}}_{\text{large}}$ | | | | | | |
| + weighted | 93.2 | 93.3 | 74.1 | 78.1 | 90.7 | 56.6 |
| + random | 92.5 | 93.1 | 73.5 | 77.5 | 90.5 | 56.0 |

Table 4: Few-Shot Performance on Test Set.

| | MRPC (F1) | SST-2 (Acc) | RTE (Acc) | BoolQ (Acc) | CB (F1) | COPA (Acc) |
|---|---|---|---|---|---|---|
| $\text{BERT}^{\text{F}}_{\text{base}}$ | 81.8 | 60.3 | 85.9 | 62.2 | 68.9 | 58.0 |
| DistilBert | 81.9 | 58.8 | 75.9 | 62.2 | 52.3 | 54.0 |
| TinyBert | 86.3 | 68.0 | 83.3 | 64.2 | 75.6 | 61.0 |
| MiniLm | 83.1 | 60.0 | 77.1 | 62.2 | 69.3 | 55.0 |
| $\text{XDT}_{\text{L6-H256-A8}}$ | 84.6 | 73.0 | 78.7 | 62.2 | 67.2 | 54.0 |
| $\text{XDT}_{\text{L6-H384-A12}}$ | 85.4 | 77.0 | 83.0 | 63.1 | 74.8 | 56.0 |
| $\text{ESEAD}^{\text{F}}_{\text{L6-H256-A8}}$ | | | | | | |
| + weighted | 86.0 | 74.0 | 82.0 | 63.3 | 73.1 | 59.0 |
| + random | 86.0 | 73.9 | 81.8 | 63.3 | 73.1 | 59.0 |
| $\text{ESEAD}^{\text{F}}_{\text{L6-H384-A12}}$ | | | | | | |
| + weighted | 87.6 | 77.6 | 83.8 | 64.5 | 77.4 | 63.0 |
| + random | 87.3 | 77.3 | 83.8 | 64.0 | 77.4 | 63.0 |

BERT among most of the tasks. The reason behind may be that TinyBERT is two-stage distilled with large amount of augmented data, so it may provide smooth generalization. As a comparison, the larger version of ESEAD performs the best over all tasks, and the improvements are between 0.6 to 7% to its initialization. The smaller version of ESEAD also improves, and the performance is improved about 1.0 to 5.9%.

Table 5: Ablation Analysis

| | MRPC (F1) | SST2 (Acc) | RTE (Acc) | BoolQ (Acc) | CB (F1) | COPA (Acc) |
|---|---|---|---|---|---|---|
| $\text{XDT}_{\text{L6-H256-A8}}$ | 90.0 | 91.2 | 78.7† | 75.4† | 70.0† | 60.0† |
| $\text{XDT}_{\text{L6-H384-A12}}$ | 90.0 | 92.3 | 80.9 | 76.8† | 81.8† | 64.0† |
| $\text{Baseline}_{\text{L6-H256-A8}}$ | 90.7 | 91.4 | 78.0 | 75.9 | 71.2 | 62.5 |
| + MLD | 92.1(1.3↑) | 91.5 (0.1↑) | 78.2 (0.2↑) | 76.0 (0.1↑) | 75.6 (4.4↑) | 64.0 (1.5↑) |
| + I.O (t=0.7) | 91.7 (1.0↑) | 91.3 (0.1↓) | 76.2 (1.8↓) | 75.7 (0.2↓) | 71.4 (0.2↑) | 63.0 (0.5↑) |
| + I.O (t=0.9) | 92.0 (1.3↑) | 91.4 | 77.3 (0.7↓) | 75.4 (0.5↓) | 71.6 (0.4↑) | 63.0 (1.0↑) |
| + R.O | 92.3 (1.7↑) | 91.9 (0.5↑) | 78.3 (0.3↑) | 76.2 (0.3↑) | 75.6 (4.4↑) | 64.0 (1.5↑) |
| + MLD + I.O (t=0.7) | 91.9 (1.2↑) | 91.9 (0.7↑) | 77.6 (0.4↓) | 75.7 (0.2↓) | 71.2 | 64.0 (1.5↑) |
| + MLD + I.O (t=0.9) | 92.2 (1.5↑) | 92.0 (0.6↑) | 77.3 (0.7↓) | 75.4 (0.5↓) | 71.4 (0.2↑) | 64.0 (1.5↑) |
| + MLD + R.O (**ESEAD**) | 92.6 (1.9↑) | 92.0 (0.6↑) | 79.4 (1.4↑) | 77.1 (1.2↑) | 89.4 (18.2↑) | 67.0 (1.0↑) |
| $\text{Baseline}_{\text{L6-H384-A12}}$ | 92.4 | 92.2 | 81.0 | 77.0 | 81.5 | 65.7 |
| + MLD | 92.5 (0.1↑) | 92.3 (0.1↑) | 81.0 (3.0↑) | 77.6 (1.1↑) | 81.8 (0.3↑) | 66.0 (0.3↑) |
| + I.O (t=0.7) | 91.7 (0.7↓) | 91.6 (0.6↓) | 80.9 (0.1↓) | 77.0 | 77.4 (4.1↓) | 64.0 (1.7↓) |
| + I.O (t=0.9) | 91.6 (0.8↓) | 91.5 (0.7↓) | 81.0 | 77.2 (0.2↑) | 76.9 (4.6↓) | 64.0 (0.4↓) |
| + R.O | 92.7 (0.3↑) | 92.5(0.3↑) | 81.9 (0.9↑) | 78.0 (1.0↑) | 86.5 (5.0↑) | 66.0 (0.3↑) |
| + MLD + I.O (t=0.7) | 92.3 (0.1↓) | 91.7 (0.5↓) | 81.0 | 77.4 (0.4↑) | 78.4 (3.1↓) | 65.0 (0.7↓) |
| + MLD + I.O (t=0.9) | 91.9 (0.5↓) | 92.0 (0.2↓) | 81.0 | 77.5 (0.5↑) | 81.2 (0.3↓) | 65.0 (0.7↓) |
| + MLD + R.O (**ESEAD**) | 93.9 (1.5↑) | 93.2 (1.0↑) | 82.4 (1.4↑) | 78.3 (1.3↑) | 93.1 (11.6↑) | 68.4 (2.4↑) |

### 3.4 ABLATION STUDY

In this section, we investigate the effectiveness of our proposed technique. We start with the baseline performance where the naive multi-teacher ensemble is applied. The naive multi-teacher ensemble takes the average of three teachers' logits as the final logits for the student model to learn. We add individual technique on top of the baseline and obtain its evaluation results. In the end, both techniques are added to compare. According to Table 5, we find that 1) Regardless of model size, the multi-logits dropout (MLD) improves performance over the baseline in all settings, and the improvement ranges from 0.1 to 4.4%. This is consistent with the point 4 in the benchmark evaluations. 2) As with MLD, random overlooking (R.O) consistently improves performance by a greater margin than MLD, from 0.3 to 5.0%. The random overlook is performed at every epoch, so the student model has the opportunity to view the data from different perspectives. For example, for the same data, the student model is randomly selected to receive knowledge from the teacher at a different epoch. 3) Informative overlooking helps, but it depends on the data. It can sometimes be detrimental to the results, possibly because students are biased to accept only the correct signal (confident knowledge) thus leading to over-fitting. 4) Both combinations e.g., MLD + R.O and MLD + I.O can produce more gains in the scores. Among some tasks, there are more gains than the increase of sum of improvement. Last but not least, Figure 2 shows that our method is beneficial for training with respect to the optimization and speed. The baseline models converges to loss at 0.28 around 1500 steps, and ours converges to a lower minima 0.22 around 1380 steps.

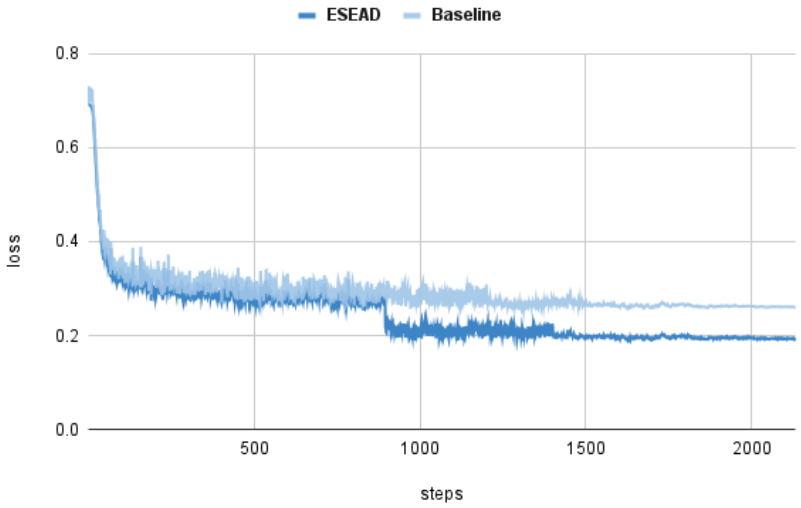



Figure 2: Comparison on Validation Loss over SST-2

## 4 DISCUSSION

In this paper, we further discuss knowledge distillation and propose a logits-based approach, called ESEAD, to improve student performance. We propose two techniques, i.e., overlooking and multi-logit dropout. One controls whether the teacher's knowledge should be received at a certain step, and the other creates an enhanced version of the teacher's logits. Numerous experiments have shown that the student model distilled by ESEAD not only outperforms feature-based approaches in generalization, but also possesses better few-shot learning. In addition, ESEAD eases the use of teaching assistants, allowing small student models with large capacity gaps to be distilled directly from large PLMs. Due to the nature of the logits-based approach, ESEAD is not limited to natural language processing only, other applications such as computer vision, audio processing, and recommender systems can be also beneficial. We leave this to future work.

## REFERENCES

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. The fifth pascal recognizing textual entailment challenge. In *TAC*, 2009.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pp. 1–14, Vancouver, Canada, 2017. Association for Computational Linguistics. doi: 10.18653/v1/S17-2001. URL https://aclanthology.org/S17-2001.

Cheng Chen, Yichun Yin, Lifeng Shang, Zhi Wang, Xin Jiang, Xiao Chen, and Qun Liu. Extract then distill: Efficient and effective task-agnostic bert distillation, 2021.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

Marie-Catherine De Marneff, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. *proceedings of Sinn und Bedeutung 23*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models, 2022a. URL https://arxiv.org/abs/2203.06904.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, Jing Yi, Weilin Zhao, Xiaozhi Wang, Zhiyuan Liu, Hai-Tao Zheng, Jianfei Chen, Yang Liu, Jie Tang, Juanzi Li, and Maosong Sun. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. 2022b.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://aclanthology.org/I05-5002.

Jia Guo, Minghao Chen, Yao Hu, Chen Zhu, Xiaofei He, and Deng Cai. Reducing the teacher-student gap via spherical knowledge disitllation. 2020.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 2019. URL http://proceedings.mlr.press/v97/houlsby19a.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Hiroshi Inoue. Multi-sample dropout for accelerated training and better generalization, 2019. URL https://arxiv.org/abs/1905.09788.

Shankar Iyer, Nikhil Dandekar, and Kornel Csernai. First quora dataset release: Question pairs, 2017. URL https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs.

Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, and Ali Ghodsi. Annealing knowledge distillation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 2493–2504, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.212. URL https://aclanthology.org/2021.eacl-main.212.

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 *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.372. URL https://aclanthology.org/2020.findings-emnlp.372.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*, 2018.

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. 2019.

Yuang Liu, Wei Zhang, and Jun Wang. Adaptive multi-teacher multi-level knowledge distillation. *ArXiv preprint*, abs/2103.04062, 2021. URL https://arxiv.org/abs/2103.04062.

Moyan Mei and Rohit Sroch. Sead: Simple ensemble and knowledge distillation framework for natural language understanding. *Lattice, THE MACHINE LEARNING JOURNAL by Association of Data Scientists*, 3(1), 2022. URL www.adasci.org/journals/lattice-35309407/?volumes=true&open=621a3b18edc4364e8a96cb63.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 5191–5198. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/5963.

Subhabrata Mukherjee, Ahmed Hassan Awadallah, and Jianfeng Gao. Xtremedistiltransformers: Task transfer for task-agnostic distillation, 2021.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 46–54, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.7. URL https://aclanthology.org/2020.emnlp-demos.7.

Mohammad Taher Pilehvar and os'e Camacho-Collados. Wic: 10, 000 example pairs for evaluating context-sensitive representations. *ArXiv preprint*, abs/1808.09121, 2018. URL https://arxiv.org/abs/1808.09121.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL https://aclanthology.org/D16-1264.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI Spring Symposium Series*, 2011.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1631–1642, Seattle, Washington, USA, 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1170.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 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)*, pp. 4323–4332, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1441. URL https://aclanthology.org/D19-1441.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 3261–3275, 2019a. URL https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b. URL https://openreview.net/forum?id=rJ4km2R5t7.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641, 2019. doi: 10.1162/tacl_a_00290. URL https://aclanthology.org/Q19-1040.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 1112–1122, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://aclanthology.org/N18-1101.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6.

Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. One teacher is enough? pre-trained language model distillation from multiple teachers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 4408–4413, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.387. URL https://aclanthology.org/2021.findings-acl.387.

Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models, 2021. URL https://arxiv.org/abs/2106.10199.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q. Weinberger, and Yoav Artzi. Revisiting few-sample BERT fine-tuning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=cO1IH43yUF.

# 5 APPENDIX

Table 6: Extended Few-Shot Performance. Results are obtained on the test set of tasks.

| | MRPC (F1) | SST2 (Acc) | RTE (Acc) | BoolQ (Acc) | CB (F1) | COPA (Acc) |
|---|---|---|---|---|---|---|
| $\text{BERT}^{\text{F}}_{\texttt{base}}$ | 70.1/81.8 | 60.3 | 85.9 | 62.2 | 83.9/68.9 | 58.0 |
| $\text{BERT}^{\text{L}}_{\texttt{base}}$ | 68.9/81.5 | 60.0 | 82.2 | 62.2 | 73.2/67.2 | 57.0 |
| $\text{BERT}^{\text{A}}_{\texttt{base}}$ | 68.9/81.5 | 60.0 | 78.9 | 62.2 | 73.2/51.5 | 57.0 |
| $\text{BERT}^{\text{B}}_{\texttt{base}}$ | 68.4/81.2 | 59.5 | 70.0 | 62.1 | 66.1/46.1 | 57.0 |
| DistilBert | 70.0/81.9 | 58.8 | 75.9 | 62.2 | 75.2/52.3 | 54.0 |
| TinyBert | 79.4/86.3 | 68.0 | 83.3 | 64.2 | 85.7/75.6 | 61.0 |
| MiniLm | 74.5/83.1 | 60.0 | 77.1 | 62.2 | 76.8/69.3 | 55.0 |
| $\text{XtremeDistilTF}_{\texttt{L6-H256-A8}}$ | 76.5/84.6 | 73.0 | 78.7 | 62.2 | 82.1/67.2 | 54.0 |
| $\text{XtremeDistilTF}_{\texttt{L6-H384-A12}}$ | 79.2/85.4 | 77.0 | 83.0 | 63.1 | 83.9/74.8 | 56.0 |
| $\text{ESEAD}^{\text{F}}_{\texttt{L6-H256-A8}}$ | | | | | | |
| + weighted | 79.2/86.0 | 73.6 | 81.8 | 63.3 | 83.9/73.1 | 59.0 |
| + random | 79.2/86.0 | 73.6 | 81.7 | 63.3 | 83.9/73.1 | 59.0 |
| $\text{ESEAD}^{\text{A}}_{\texttt{L6-H256-A8}}$ | | | | | | |
| + weighted | 78.4/86.0 | 73.6 | 80.8 | 63.1 | 83.9/68.9 | 59.0 |
| + random | 78.4/86.0 | 73.6 | 80.8 | 63.0 | 83.9/68.9 | 58.0 |
| $\text{ESEAD}^{\text{L}}_{\texttt{L6-H256-A8}}$ | | | | | | |
| + weighted | 78.4/86.0 | 73.6 | 80.6 | 63.1 | 83.9/68.9 | 59.0 |
| + random | 78.4/86.0 | 73.3 | 80.6 | 63.1 | 83.9/68.9 | 59.0 |
| $\text{ESEAD}^{\text{B}}_{\texttt{L6-H256-A8}}$ | | | | | | |
| + weighted | 78.2/85.8 | 73.3 | 80.4 | 63.0 | 83.9/67.9 | 56.0 |
| + random | 77.5/85.3 | 73.3 | 80.4 | 62.6 | 83.9/67.9 | 56.0 |
| $\text{ESEAD}^{\text{F}}_{\texttt{L6-H384-A12}}$ | | | | | | |
| + weighted | 81.1/87.3 | 77.3 | 83.5 | 64.3 | 85.7/77.4 | 63.0 |
| + random | 80.9/87.1 | 77.3 | 83.5 | 63.8 | 85.7/77.4 | 63.0 |
| $\text{ESEAD}^{\text{L}}_{\texttt{L6-H384-A12}}$ | | | | | | |
| + weighted | 80.6/86.9 | 77.3 | 83.5 | 63.8 | 85.7/76.1 | 63.0 |
| + random | 80.1/86.3 | 77.2 | 83.3 | 63.8 | 85.7/76.1 | 60.0 |
| $\text{ESEAD}^{\text{A}}_{\texttt{L6-H384-A12}}$ | | | | | | |
| + weighted | 80.9/87.0 | 77.3 | 83.3 | 63.8 | 85.7/75.6 | 60.0 |
| + random | 80.1/86.3 | 77.3 | 83.3 | 63.8 | 85.7/75.6 | 60.0 |
| $\text{ESEAD}^{\text{B}}_{\texttt{L6-H384-A12}}$ | | | | | | |
| + weighted | 80.4/86.7 | 77.2 | 83.1 | 63.4 | 83.9/74.8 | 59.0 |
| + random | 80.1/86.3 | 77.2 | 83.0 | 63.3 | 83.9/74.8 | 58.0 |

Table 7: ESEAD HyperParameters for GLUE and SuperGlue

| Hyper-Parameters | Choices |
|---|---|
| Seed | {1, 42, 88} |
| Batch Size | {16, 32} |
| Weight Decay | {0.0, 0.1} |
| Max Epochs | 20 |
| Warm-up Ratio | 0.1 |
| KD Loss Type | {MSE, KL} |
| Overlook $r$ | {0.1, 0.2} |
| Multi-Logits Dropout $r$ | {0.1, 0.2} |
| Multi-Logits Dropout $d$ | 20 |
| LR | {2e-5, 3e-5} |
| LR scheduler | linear |
| Temperature | 20 |
| Adam $\epsilon$ | 1e-6 |
| Adam $\beta_1$ | 0.9 |
| Adam $\beta_2$ | 0.98 |