# Adversarial Data Augmentation for Task-Specific Knowledge Distillation of Pre-Trained Transformers

**Minjia Zhng, Niranjan Uma Naresh, Yuxiong He**

Microsoft Corporation
Bellevue, Washington 98004
{minjiaz,Niranjan.Uma,yuxhe}@microsoft.com

## Abstract

Deep and large pre-trained language models (e.g., BERT, GPT-3) are state-of-the-art for various natural language processing tasks. However, the huge size of these models brings challenges to fine-tuning and online deployment due to latency and cost constraints. Existing knowledge distillation methods reduce the model size, but they may encounter difficulties transferring knowledge from the teacher model to the student model due to the limited data from the downstream tasks. In this work, we propose $AD^2$, a novel and effective data augmentation approach to improve the task-specific knowledge transfer when compressing large pre-trained transformer models. Different from prior methods, $AD^2$ performs distillation by using an enhanced training set that contains both original inputs and adversarially perturbed samples that mimic the output distribution from the teacher. Experimental results show that this method allows better transfer of knowledge from the teacher to the student during distillation, producing student models that retain 99.6% accuracy of the teacher model while outperforming existing task-specific knowledge distillation baselines by 1.2 points on average over a variety of natural language understanding tasks. Moreover, compared with alternative data augmentation methods, such as text-editing-based approaches, $AD^2$ is up to 28 times faster while achieving comparable or higher accuracy. In addition, when $AD^2$ is combined with more advanced task-agnostic distillation, we can advance the state-of-the-art performance even more. On top of the encouraging performance, this paper also provides thorough ablation studies and analysis. The discovered interplay between KD and adversarial data augmentation for compressing pre-trained Transformers may further inspire more advanced KD algorithms for compressing even larger scale models.

## 1 Introduction

There has been a huge paradigm shift in AI: large-scale foundation models (Bommasani et al. 2021), such as BERT (Devlin et al. 2019) and GPT-3 (Brown et al. 2020), are trained on massive data at scale and then are adapted to a wide range of different domains with additional task-specific data. One interesting trend of these foundation models is their sizes grow at an unprecedented speed, from a few hundred million parameters (e.g., BERT) to over one hundred billion parameters (e.g., GPT-3), a three-orders-of-magnitude increase. Recent studies from OpenAI have shown that the model scale has been increasing exponentially with roughly a 3.4-month doubling time and the performance of these models continues to improve with their sizes (Kaplan et al. 2020). Despite their remarkable performance in accuracy, huge challenges have been raised when deploying applications on top of these foundation models (e.g., efficient inference) due to latency and capacity constraints.

One effective approach for reducing the model size is knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015), where a stronger model (called teacher) guides the learning of another small model (called student) with an objective to minimize the discrepancy between the teacher and student outputs. Since its debut, KD has been extensively applied to computer vision and NLP tasks (Wang and Yoon 2020; Gou et al. 2021). On the NLP side, several variants of KD have been proposed to compress BERT (Devlin et al. 2019), including how to define the knowledge that is supposed to be transferred from the teacher BERT model to the student variations. Examples of such knowledge definitions include output logits (e.g., DistilBERT (Sanh et al. 2019)) and intermediate knowledge such as feature maps (Sun et al. 2019; Aguilar et al. 2020; Zhao et al. 2021) and self-attention maps (Wang et al. 2020b; Sun et al. 2020) (we refer KD using these additional knowledge as deep knowledge distillation (Wang et al. 2020b)). Unfortunately, the gap between the teacher and the student is sometimes large, even with deep distillation, especially when the downstream task data is limited. To mitigate the accuracy gap, existing work has explored applying knowledge distillation in the more expensive pre-training stage, which aims to provide a better initialization to the student for adapting to downstream tasks. As an example, MiniLM (Wang et al. 2020b) and MobileBERT (Sun et al. 2020) advance the state-of-the-art by applying deep knowledge distillation and architecture change to pre-train a student model on the general-domain corpus, which then can be directly fine-tuned on downstream tasks with good accuracy. TinyBERT (Jiao et al. 2019) proposes to perform deep distillation in both the pre-training and fine-tuning stage and shows that these two stages of knowledge distillation are complementary to each other and can be combined to achieve state-of-the-art results on GLUE tasks.

Despite the great progress, there remains one big challenge — while the pre-training distillation may benefit from

unsupervised learning over a huge amount of general domain data, the fine-tuning distillation often has only limited labeled data from the target domain for a certain task. Indeed, large amounts of labeled data are usually prohibitive or expensive to obtain. Due to the limited data from the target task/domain, fine-tuning distillation can cause the adapted model overfit the training data and therefore does not generalize well. Existing methods such as TinyBERT try to overcome this issue by enlarging the training datasets with text-editing-based data augmentation. However, such a method requires generating multiple samples for each input to have adequate variations, which leads to excessive data augmentation time and drastically increases the training cost. In this paper, we will show that instead of doing text-editing data augmentation, we can achieve better distillation performance on low-resource downstream tasks with much cheaper cost by the original KD loss combined with a strong and more principled adversarial data augmentation scheme.

**Our Contributions.** (1) We introduce AD$^2$, a novel task-specific knowledge distillation method for compressing pre-trained Transformer networks via a strong adversarial data augmentation scheme. (2) We conduct a comprehensive comparison of AD$^2$ with prior task-specific KD methods on a wide range of NLP tasks and demonstrate that AD$^2$ teaches a student to get little or no loss in accuracy and retains 99.6% accuracy of the teacher model on average over GLUE benchmark, outperforming existing task-specific knowledge distillation methods by 1.2 points in accuracy under the same compression ratio. (3) We also perform a comparison with an existing text-edit-based data augmentation (DA) method used in TinyBERT (Jiao et al. 2019). Our results show that AD$^2$ achieves better accuracy than DA while being 8.6–28 times faster than DA. (4) We show that our approach complements existing pre-training distillation and enhances state-of-the-art pre-training distillation methods to advance the state-of-the-art further (e.g., up to 1.1 points higher accuracy when combined with MiniLM). (5) We perform detailed ablation studies to assess the impact of our method.

## 2 Background and Related Work

Knowledge distillation (KD) was first introduced by (Bucila, Caruana, and Niculescu-Mizil 2006) and later generalized by (Hinton, Vinyals, and Dean 2015). It has been demonstrated as an empirically very successful technique. In particular, the challenge of deploying large-scale pre-trained Transformer-based language models (e.g., BERT) motivates many works to improve the knowledge distillation technique for Transformer models, by exploiting additional intermediate knowledge (Sun et al. 2019; Aguilar et al. 2020; Tang et al. 2019), task-agnostic pre-training distillation (Wang et al. 2020b; Sun et al. 2020), and multi-stage distillation (Wang et al. 2020b; Jiao et al. 2019). In contrast to previous works, our goal in this paper is to improve the task-specific KD performance for pre-trained Transformer models where downstream task data resources are limited with the help of adversarial data augmentation. We empirically show that this path is effective while being lightweight.

On a separate line of research, data augmentation has been a prevailing technique to overcome overfitting and improve generalization. For example, in image classification tasks, data augmentation applies label-invariant transformations (such as cropping, flipping, color jittering) to images in the training so that the model can learn representations robust to those nuisance factors. Text data augmentation has also been extensively studied in NLP. For example, prior work proposes to improve neural machine translation models with back-translation (Sennrich, Haddow, and Birch 2016). Other work propose to replace words with other words that are predicted using a language model at the corresponding word positions (Kobayashi 2018; Wu et al. 2019). EDA proposes to augment text data through synonym replacement, random swap, random insertion, and random deletion, which shows improved performance on text classification tasks using LSTMs (Wei and Zou 2019). Unlike these methods, which focus on general data augmentation for NLP tasks, our work is the first to investigate the interplay between adversarial data augmentation and knowledge distillation loss for compression of pre-trained Transformer models, with limited data from downstream task.

The work most similar to our research is TinyBERT(Jiao et al. 2019). TinyBERT uses a text-editing technique for data augmentation by randomly replacing words in a sentence with their synonyms, based on their similarity measure on GloVe embeddings. It then uses the augmented dataset for task-specific distillation of BERT models. While this kind of augmentation can keep semantics at the word level, it still has one big limitation: to generate new sentence samples with adequate variations, it needs to sample multiple times. For example, to achieve improved accuracy, the data augmentation used by (Jiao et al. 2019) increases training sets by a factor of 10–30, which not only leads to high augmentation cost but also increases the distillation training cost by close to an order of magnitude. Unlike (Jiao et al. 2019), we consider a worst-case formulation over data distributions and propose an adversarial data augmentation method for distillation, which results in better improvement while incurring a much cheaper cost on distilling task-specific student models.

## 3 Proposed Method

### 3.1 Problem Statement

Consider a pre-trained large-scale language model $T_\Theta(\cdot)$ for adapting to natural language understanding tasks, such as sentiment analysis, question and answering and semantic textual similarity, where the labeled input data is $\{x_i, y_i\}_{i=1}^N$; $x_i$ represents the $i^{\text{th}}$ input (typically sentences) where a special token $[SEP]$ is used to indicate the sentence boundary and a $[CLS]$ symbol appended to the front of the input used for tasks such as classification, and $y_i$ is the corresponding ground-truth label. In the standard fine-tuning framework, the model $T$ is initialized with pre-trained parameters $\Theta_0$ and fine-tuned with labeled data by minimizing the task-specific objective (e.g., cross-entropy for classification tasks): $\min_\Theta \mathbb{E}_{(x,y)\sim D}\mathcal{L}_{ce}(\Theta) = \min_\Theta \sum_{i=1}^N \sum_{j=1}^{|y_i|} y_{ij} \log(\text{softmax}(T_\Theta(x_i))_j)$. The large-scale (teacher) model has unwieldy computation and memory requirements. Therefore, the goal is to learn a task-specific

smaller model, $S$ (parameterized by $\theta$), without degrading the accuracy in comparison to the teacher model $T_\Theta$.

### 3.2 Adversarial Training

Adversarial training has been proposed and studied extensively in the computer vision literature mainly for improving model robustness and withstand adversarial attacks (Goodfellow, Shlens, and Szegedy 2015; Madry et al. 2018). The key idea is to apply small perturbation to input images that maximizes the adversarial loss:

$$\min_\theta \mathbb{E}_{(x,y)\sim D}[\max_{\|\delta\|\leq\epsilon} l(f(x+\delta;\theta),y)] \quad (1)$$

While adversarial training has been successfully mitigating adversarial attacks, traditional understanding is that adversarial training could hurt generalization performance. However, there has been an increasing amount of attention paid to leverage adversarial training for better clean data performance (Xie et al. 2020; Zhu et al. 2020; Gan et al. 2020). In particular, there are some studies show that adversarial training helps improve the generalizability of language modeling (Cheng, Jiang, and Macherey 2019; Wang, Gong, and Liu 2019; Jiang et al. 2020; Liu et al. 2020). However, few works have studied its interplay with knowledge distillation. Given that both could improve model generalization, it poses the question: to what extend does distillation of pre-trained Transformers benefit from task-specific adversarial training?

### 3.3 The AD$^2$ Algorithm

Adversarial training is a form of data augmentation. In this section, we introduce AD$^2$, **A**dversarial **D**ata **A**ugmentation for **D**istillation (AD$^2$), to exploit adversarial data augmentation techniques for knowledge distillation.

Due to the intrinsic difference between the image and text data, adversarial training methods for images cannot be directly applied to the latter one. First, the image (e.g., pixels) is continuous-valued, but text data is discrete. In a pre-trained language model, raw text data is first vectorized, such as one-hot encoding, before getting fed into the model. When applying gradient-based adversarial training method adopted from images on these representations, the generated adversarial samples contain invalid tokens or word sequences (Alzantot et al. 2018). Second, while the perturbation of images is small changes of pixel values that are hard to be perceived by human eyes and label-preserving, word replacement (e.g., even with synonyms) would generate syntactically incorrect sentences and may change the semantics of a sentence drastically (Jia and Liang 2017).

To address this issue, we create adversarial samples by applying perturbations to the continuous lexical embeddings of inputs to the student model instead of directly to discrete words or tokens. Using BERT as an example, for a raw input $x_i$, we pass it to the lexical encoding layer of the student model $g_{emb}(x_i) = \text{LexicalEncoder}(x_i)$, which combines a token embedding (e.g., one-hot), a position embedding, and a segment embedding through element-wise summation. Then we add $g_{emb}(x_i)$ with a vector

$$\delta_\theta = \arg\max_{\|\delta\|\leq\epsilon} \phi(S_\theta(g_{emb}(x_i)+\delta), T_\Theta(x_i)) \quad (2)$$

which represents the worst-case perturbations against the teacher model's outputs. We choose the teacher model's output instead of the hard labels for two reasons. First, a data augmentation scheme has to supply corresponding labels as supervisory information. Therefore, our data augmentation scheme needs not to worry about the labels as they are assigned by the teacher model. Second, the teacher's output provides richer information about the relationship between samples. Therefore, we consider it a better reference point for the adversarial direction, which is the direction in the input space in which the label probability of the model is most sensitive to small perturbations. Finally, from a semantic-preserving perspective, we cannot perform very "extreme" transformations for data augmentation. Therefore, we restrict the magnitude of the perturbation to $\epsilon$ (e.g., by simple clipping), such that the perturbation lies within an $L_2$-norm ball with a radius of $\epsilon$. For optimization, Equation 2 can be solved by project gradient ascent (PGA) (Madry et al. 2018), which is commonly used for large-scale constrained optimization.

After we generate adversarial samples, unlike common data augmentation where only the transformed inputs are fed into the network, we pass both the original input $x_i$ and the adversarial sample $x_i' = g_{emb}(x_i) + \delta$ for training (thus, the number of input samples during training is increased by a factor of 2). The consideration of keeping both inputs is to maintain the information path for the original input $x_i$ so that we can easily see how the added information path $x'$ leads to a different result. Figure 1 shows how AD$^2$ applies adversarial token embeddings to the student and how they are used during distillation.

For the $x_i$ part, its loss is still the original KD loss (i.e., $\mathcal{L}_{KD}$), which is a weighted sum of (1) the conventional cross-entropy loss between predictions and the given hard label and (2) the Kullback–Leibler divergence (KL) loss between the predictions and the teacher's soft label. For the $x_i'$ part, we use the KL divergence to calculate the adversarial data augmentation loss (ADA), i.e., $\mathcal{L}_{ADA}(x_i,\delta;\theta) := \text{KL}(S_\theta(g_{emb}(x_i)+\delta), T_\Theta(x_i))$ for classification tasks. For regression tasks, both $S$ and $T$ output a scalar, and we set $\mathcal{L}_{ADA}$ as the squared loss, i.e., $\mathcal{L}_{ADA}(x_i,\delta;\theta) := (S_\theta(g_{emb}(x_i)+\delta) - T_\Theta(x_i))^2$. Thus, our approach encourages a student network to produce the softmax output from the teacher network when exposed to adversarial samples, i.e., to minimize the following objective:

$$\min_\theta \mathbb{E}_{(x,y)\sim D}[\mathcal{L}_{KD}(x;\theta) + \alpha\mathcal{L}_{AD^2}(x';\theta)] \quad (3)$$

where $\alpha$ is a hyperparameter that controls the trade-off between KD loss from original data and adversarial data. In our experiments, we set $\alpha = 1$ except otherwise noted.

**Computational cost of augmentation.** Given that one primary interest is to also improve the distillation efficiency, we are motivated to also look into the computational cost of adversarial data augmentation. Generating adversarial examples requires K PGA iterations, where each iteration takes approximately the same time as making three forward passes through the network. This is because one step of PGA requires to make one forward and backward pass over the entire network. When K is large, the data augmentation cost can still be expensive. Inspired by prior work on reducing the

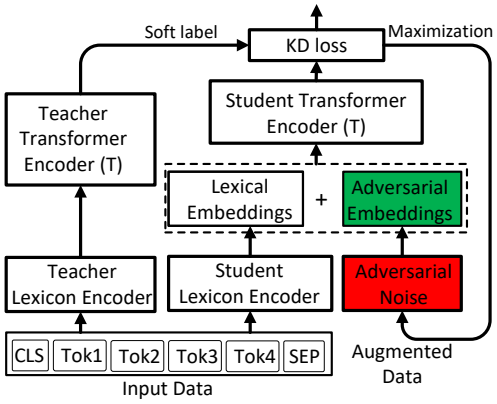


Figure 1: AD$^2$ architecture. A text input is first fed to a teacher model to generate the soft labels. AD$^2$ then creates adversarial samples by applying perturbations to the continuous lexical embeddings of inputs. Both the original inputs and the adversarial samples are passed to the student model for knowledge distillation.

training cost of adversarial training, we employ a variant of PGA called PGA-1 (Gupta, Dube, and Verma 2020) (K=1) to craft adversarial samples with one perturbation step. The key insight here is that we can often find sufficiently good adversarial samples while being much more computationally efficient with small K for many tasks. More advanced adversarial training approaches, such as Curriculum Adversarial Training (CAT) (Cai, Liu, and Song 2018) and Annealing-based Adversarial Training (Amata) (Ye et al. 2020), may help further reduce the adversarial training cost while maintaining good accuracy by adjusting the strength of adversaries at different stages of the training. We leave the exploration of these methods as future work.

**Practical considerations.** Since we study the effectiveness of adversarial data augmentation for KD, we still use a common temperature term $t$ to control how much to rely on the teacher's soft predictions, where we divide the logits of both student and teacher by $t$ (e.g., $\widehat{y}_i = P(y_i/t|x_i)$) for both raw data and adversarial samples during distillation. A high temperature has the effect of generating a softer distribution of output probabilities among the classes (Hinton, Vinyals, and Dean 2015). And we scale the gradient of the loss with respect to the model weights by a factor of $t^2$ such that the relative contributions of the loss term remain roughly unchanged even when the temperature is adjusted.

Inspired by MobileBERT (Sun et al. 2020), which performs deep knowledge distillation of the teacher model during the pre-training distillation, we let the student imitate the prediction output, feature maps, and self-attention maps with adversarial data augmentation at the task-specific distillation stage. We show that adversarial data augmentation is beneficial to KD with and without deep knowledge distillation, but they can be combined together to deliver better results. The full procedure of AD$^2$ is provided in Algorithm 1.

Algorithm 1: **AD$^2$**

**Input:** Teacher network $T$, coefficient $\alpha$, temperature $t$, maximum perturbation radius $\epsilon$.
**Output**: Converged model parameters $\theta$ of the distilled student network $S$
**for** $epoch \in 1, 2, ..., N_{epochs}$ **do**
  **for** Each $(x, y) \in$ mini-batch $(X, Y) \sim D$ **do**
    $\widehat{y} \leftarrow T^t_{\Theta}(x)$ ▷ Calculate the soft labels from the teacher
    $\mathcal{L}_{CE} \leftarrow CE(S_\theta(x), y)$ ▷ Calculate the standard loss.
    $\mathcal{L}_{KD} \leftarrow KL(S^t_\theta(x), \widehat{y})$ ▷ Calculate the KD loss on clean data
    $\delta_\theta = \arg\max_{\|\delta\| \leq \epsilon} \phi(S_\theta(g_{emb}(x) + \delta), \widehat{y})$ ▷ Compute data perturbation
    $x' = g_{emb}(x) + \delta_\theta$ ▷ Create adversarial sample
    $\mathcal{L}_{ADA} \leftarrow KL(S^t_\theta(x'), \widehat{y})$ ▷ Calculate the KD loss on adversarial data
    $\mathcal{L}_{AD^2} \leftarrow \mathcal{L}_{CE} + \mathcal{L}_{KD} + \alpha\mathcal{L}_{ADA}$ ▷ Calculate the final loss
    Update student model parameters

# 4 Experiments

In this section, we describe our experiments on the proposed adversarial data augmentation for knowledge distillation.

## 4.1 Evaluation Methodology

**Datasets.** Following previous work on distilling pre-trained language model (Sanh et al. 2019; Sun et al. 2019; Dong et al. 2019), we evaluate the effectiveness of AD$^2$ using the GLUE (General Language Understanding Evaluation) benchmark (Wang et al. 2019), a collection of linguistic tasks in different domains such as textual entailment, sentiment analysis, and question answering. It is designed to favor sample-efficient learning and knowledge transfer across a range of different linguistic tasks in different domains.

**Experimental settings.** We focus our comparison under a task-specific compression setting (Sun et al. 2019; Turc et al. 2019) instead of a pretraining distillation setting (Sanh et al. 2019; Wang et al. 2020b; Sun et al. 2020). That is, we do not use the massive general domain corpus but only the training set of each task in GLUE to compress the model. The reason is that we intend to straightforwardly verify the effectiveness of our adversarial data augmentation-based distillation method. Moreover, our motivation comes from improving compression efficiency for low-resource downstream tasks (e.g., no longer than a few GPU hours for any task of GLUE). If task-specific distillation already provides satisfactory accuracy, one may be spared from exploring more time-consuming pre-training distillation schemes (e.g., pre-training distillation-based methods, such as MobileBERT (Sun et al. 2020) and MiniLM (Wang et al. 2020b), take hundreds of GPU hours to obtain improved accuracy). That said, we will show that our method can be combined with pre-training distillation to achieve better results.

**Implementation Details.** Previous work (Sanh et al. 2019; Sun et al. 2019; Wang et al. 2020b) usually distil

$\text{BERT}_{base}$ (Devlin et al. 2019) into a 6-layer ($\text{BERT}_6$) model with 768 hidden size. To make the results comparable to other work, we conduct distillation experiments using the same teacher and student architecture. We use the uncased version of $\text{BERT}_{base}$ [1] (denoted as $\text{BERT}_{12}$), which consists of 12-layer Transformer blocks with 768 hidden dimension size, and 12 attention heads, with about 109M parameters. We fine-tune $\text{BERT}_{base}$ on a downstream task as the teacher model to compute soft labels for each task independently. We initialize the student model ($\text{BERT}_6$) with model weights from DistilBERT checkpoints [2]. Note that DistilBERT still uses a pre-training distillation setting. We choose DistilBERT because it provides a better baseline for BERT-PKD than initializing with weights selected from the teacher BERT.

**Hyperparameters.** In order to reduce the hyperparameter search space, we fix the number of epochs as 6 for all the experiments and tune the batch size from $\{16, 32\}$ and learning rate from {1e-5, 3e-5, 5e-5, 7e-5, 9e-5, 1e-4} for all configurations on each task. The maximum sequence length is set to 512. We use a linear learning rate decay schedule with a warm-up ratio of 0.1 for all experiments. We clip the gradient norm within 1. For $\text{AD}^2$, we set the perturbation radius $\epsilon$ = 1e-5, PGA step size 1e-3, temperature $t$=1, and $\alpha$ = 1. The model with the best validation accuracy is selected for each task, and we report the median of 5 runs with different random seeds for each selected configuration.

## 4.2 Experimental Results

We first compare the following schemes: (1) **Fine-tune**: we directly fine-tune the student model on GLUE tasks to obtain a natural fine-tuning baseline. (2) **Vanilla KD**: This is the knowledge distillation in its purest form as in (Hinton, Vinyals, and Dean 2015) and (Sanh et al. 2019). (3) **BERT-PKD** (Sun et al. 2019): a task-specific distillation technique that exploits intermediate knowledge for better compression. (4) **$\text{AD}^2$**: This is our approach as described in Algorithm 1, using adversarial data augmentation. Table 1 shows the comparison results. For a fair comparison, we reproduce results for BERT-PKD, because several results on the GLUE development set were missed from their paper. Our reproduced results of BERT-PKD are comparable and sometimes stronger than the originally reported results in (Sun et al. 2019).

**Improving distillation accuracy with $\text{AD}^2$:** Overall, $\text{AD}^2$ retains 99.6% (83.6 vs. 83.9) of the BERT-base performance, and we have the following observations. (1) $\text{AD}^2$ consistently outperforms the fine-tuning baseline on every GLUE task and achieves 1.6 points higher accuracy (82.0 vs. 83.6), indicating that our method can effectively transfer knowledge from the teacher model to the student. (2) Comparing $\text{AD}^2$ with vanilla KD and BERT-PKD, we see the proposed scheme of adopting adversarial data augmentation for distillation improves the accuracies of all teacher-student pairs. On 5 out of the 8 pairs, the improvement is more than 1 point. Notably, we observe that $\text{AD}^2$ achieves 1.1 points higher accuracy for SST-2, 1.6 points for CoLA, 1.8/1.3 points for STS-B in PCC/SCC, 1.3 points for MRPC in accuracy, and 1.5 points for RTE. (3) $\text{AD}^2$ is particularly effective in improving KD on low-resource datasets, which contain fewer samples (e.g., <100K samples). We hypothesize that this is because $\text{AD}^2$ provides a more diverse data view via strong adversarial data augmentation, such that more of the teacher's knowledge can get exposed to the student, which is challenging when the number of samples is small. (4) We highlight that on SST-2 and MRPC, our method achieves nearly identical performance to BERT-base, and on QQP and RTE, our method even outperforms BERT-base by 0.4 points (90.8 vs. 91.2) and 4 points (64.2 vs. 68.2), respectively. This is presumably because adversarial data augmentation can effectively help prevent overfitting of the student model on downstream tasks, leading to improved generalization. (5) Finally, on both large datasets with more than 100K samples (e.g., MNLI, QQP, QNLI) and low-resource datasets, $\text{AD}^2$ achieves consistent improvements, verifying the robustness of our approach.

**Exploring alternative data augmentation schemes.** It has been show that adversarial data augmentation improves accuracy of KD. In Figure 2, we compare $\text{AD}^2$ with alternative data augmentation method. Following TinyBERT (Jiao et al. 2019), we employ its text-editing technique (e.g., synonym replacement) and use the recommended parameters listed in their code repository {$p_t$=0.4, $N_a$=[10 (MNLI, QQP), 20 (QNLI, SST-2), 30 (CoLA, MRPC, RTE)], K=15} for compressing BERT models. We notice that text-editing-based data augmentation only seems to provide sparse and inconsistent improvements on GLUE tasks (e.g., we observe worse performance on MRPC, SST2 with text-edit-based DA) for the $\text{BERT}_6$ student model, despite with several hyperparameter tuning. In those cases, we report the best accuracy we observe from fine-tuning TinyBERT checkpoint or through task-specific distillation but with clean data.

By our analysis, $\text{AD}^2$ boosts KD performance more than TinyBERT on all tasks except RTE (the text-edit-based data augmentation does bring extra accuracy improvement for RTE). Our approach provides better accuracy because the replacement-based augmentation with synonyms can only produce limited diverse patterns from the original texts, and it is almost impossible to leverage all the possible candidates due to the large vocabulary size in languages. In contrast, we consider a worst-case formulation over data distributions in the semantic space. Thus, $\text{AD}^2$ augments the dataset with examples that are "hard" under the current model. We find that such stronger data augmentation can efficiently transfer teacher knowledge to the student.

On the other hand, while it takes TinyBERT 85 hours on one NVIDIA V100 GPU to perform task-specific distillation on MNLI, it takes 5.9 hours for $\text{AD}^2$ to achieve higher accuracy (83.5 vs. 83.7). The task-specific distillation part in TinyBERT is slow because it needs to perform a kNN search over the GloVe embeddings to find synonyms for a word replacement, which can be extremely expensive for large vocabulary and high embedding dimensions. Furthermore, the data augmentation scheme in TinyBERT increases the training set by at least a factor of 10 to cover enough variations. As a result, $\text{AD}^2$ is able to achieve 0.7 points better accuracy

[1] https://huggingface.co/bert-base-uncased.

[2] https://huggingface.co/distilbert-base-uncased.

| Model | Arch. | #Params. | MNLI-m -/mm (Acc.) 393K | QQP (F1/Acc.) 368K | QNLI (Acc.) 108K | SST-2 (Acc.) 67K | CoLA (MCC) 8.5K | STS-B (PCC/SCC) 5.7K | MRPC (F1/Acc.) 3.7K | RTE (Acc.) 2.5K | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BERT | 12L × 768H | 109M | 84.5/84.8 | 87.7/90.8 | 90.5 | 92.6 | 55.2 | 90.3/89.7 | 90.6/86.2 | 64.2 | 83.9 |
| Fine-tune | 6L × 768H | 66M | 82.4/82.5 | 87.1/90.3 | 89.1 | 90.9 | 53.4 | 85.6/85.5 | 89.6/85.0 | 63.5 | 82.0 |
| Vanilla KD | 6L × 768H | 66M | 82.9/82.8 | 87.3/90.5 | 89 | 91.3 | 52.4 | 84.7/84.7 | 90.3/86.0 | 66 | 82.3 |
| BERT-PKD | 6L × 768H | 66M | 83.2/82.9 | 87.6/90.7 | 89.1 | 91.5 | 53.1 | 84.6/84.7 | 90.0/85.2 | 66.7 | 82.4 |
| AD$^2$ | 6L × 768H | 66M | **83.7/84.1** | **88.2/91.2** | **91** | **92.6** | **54.7** | **86.4/86.0** | **90.6/86.5** | **68.2** | **83.6** |

Table 1: The evaluation results of the GLUE benchmark on the development set. The number below each task denotes the number of training examples. AD$^2$ outperforms existing task-specific knowledge distillation techniques by 1.2 points on average.

| Model | MNLI | QQP | QNLI | SST-2 | CoLA | MRPC | RTE | Avg |
|---|---|---|---|---|---|---|---|---|
| TinyBERT | 83.5 | 90.6 | 90.5 | 91.6 | 49.5 | 88.4 | **72.9** | 81.0 |
| AD$^2$ | **83.7** | **91.2** | **91** | **92.6** | **54.7** | **90.6** | 68.2 | **81.7** |

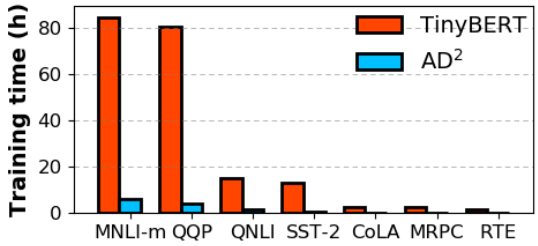


Figure 2: KD evaluation accuracy and training cost comparison when using different data augmentation schemes. TinyBERT uses text-editing based data augmentation for KD. AD$^2$ uses adversarial data augmentation for KD.

on average than TinyBERT while being 8.6–28 times faster.

**Interplay between pre-training distillation and AD$^2$.** Our approach provides additive improvements on top of state-of-the-art pre-training based distillation methods. We summarize the results in Table 2. To assess whether the gains from AD$^2$ is additive to more recent compression schemes (Wang et al. 2020b; Jiao et al. 2019; Sun et al. 2020) that perform deep knowledge distillation using general-domain data at the pre-training stage, we take a checkpoint from the latest version of MiniLM (Wang et al. 2020a), a 12-layer model with 384 hidden sizes (33M parameters) distilled from a BERT$_{base}$ size model[3]. We choose MiniLM because it achieves state-of-the-art accuracy. We take its publicly released checkpoint to initialize the student model, and we fine-tune BERT-base on each task independently as the teacher model. We then apply AD$^2$ to perform adversarial data augmentation and use those data for task-specific distillation. We exclude MobileBERT (Sun et al. 2020) in this comparison due to its redesigned Transformer block and different model size. From Table 2, we can see that AD$^2$ further enhances the performance of MiniLM. Overall, AD$^2$ consistently brings non-trivial accuracy improvement on GLUE tasks and improves the accuracy by 0.5 points on average, which demonstrates its versatility in terms of combining with pre-training distillation to further collect performance gains and advance the state-of-the-art.

[3]https://huggingface.co/microsoft/MiniLM-L12-H384-uncased

| Model | MNLI-m | QQP | QNLI | SST-2 | MRPC | RTE | Avg |
|---|---|---|---|---|---|---|---|
| MiniLM | 85.6 | 90.9 | 91.3 | 92.8 | **90.1** | 71.8 | 87.1 |
| +AD$^2$ | **86.0** | **91.4** | **91.8** | **93.1** | **90.1** | **72.9** | **87.6** |

Table 2: Evaluation results on pre-training distillation checkpoint, w/o and with AD$^2$ based task-specific distillation.

### 4.3 Analysis

In this section, we first perform an ablation study and analyze the effectiveness of AD$^2$. We then evaluate how the proposed method works with alternative initialization methods for the student model. Finally, we study how effective our method is as the teacher model evolves.

**Ablation studies.** In this section, we study the effectiveness of AD$^2$ by comparing the following schemes. **(1) AD$^2$:** This is our adversarial data augmentation based task-specific distillation technique. **(2) -DKD:** Like the above configuration but disables deep knowledge distillation, e.g., learning from the teacher's feature maps and self-attention maps. **(3) -KD:** Like the above configuration but disables knowledge distillation completely so we fine-tune the student directly with adversarial data augmentation. **(4) -AD:** We further disable adversarial data augmentation so that we directly performs task-specific fine-tuning to a student model.

The results are reported in Table 3. Overall, the removal of either component results in a performance drop. For example, removing deep knowledge distillation (i.e., *-DKD*) leads to 0.3 points lower accuracy (86.2 vs. 85.9), indicating that deep knowledge distillation is not only useful for pre-training distillation but can also bring benefits to task-specific distillation when adversarial samples are presented. Removing KD completely (i.e., *-KD*) leads to another 0.7 points of accuracy drop (85.9 vs. 85.2), indicating that KD is still crucial for transferring knowledge from the teacher to the student and adversarial data augmentation alone is not sufficient to close the generalization gap between the teacher and the student model. Finally, removing adversarial data augmentation (i.e., *-AD*) leads to a big accuracy drop (e.g., 0.9 points drop from 85.2 to 84.3), indicating that adversarial data augmentation is important to obtain stronger performance, especially for small tasks with limited data (e.g., SST-2, MRPC). These results demonstrate that these components complement each other and are important to obtain high accuracy for compressing the pre-trained Transformer models.

| Model | MNLI-m -/mm (Acc.) | QQP (F1/Acc.) | QNLI (Acc.) | SST-2 (Acc.) | MRPC (F1/Acc.) | RTE (Acc.) | Avg |
|---|---|---|---|---|---|---|---|
| $AD^2$ | **83.7/84.1** | **88.2/91.2** | **91** | **92.6** | **86.5/90.6** | **68.5** | **86.2** |
| -DKD | 83.5/83.5 | 88.1/**91.2** | **91** | 92.5 | 86.0/90.3 | 67.5 | 85.9 |
| -KD | 83.0/83.2 | 88.2/91.2 | 90.3 | 92.5 | 85.2/89.9 | 63.8 | 85.2 |
| -AD | 82.4/82.5 | 87.1/90.3 | 89.1 | 90.9 | 86.6/86.3 | 63.5 | 84.3 |

Table 3: Ablation results of $BERT_6$ student model distilled from $BERT_{12}$ on GLUE. The results show that removing either deep knowledge distillation, KD, or adversarial data augmentation hurts the accuracy of the student model.

**Exploring different model initialization schemes.** In Sec. 4.2, we evaluated how our method performs with checkpoints from pre-training distillation. However, the pre-training compression is still quite time-consuming. Recent work (Sajjad et al. 2020) proposes a lightweight way to obtain a compressed Transformer model by directly selecting a subset of pre-trained Transformer weights to form a compressed model. This method can be useful for compressing very large-scale models, such as GPT-3, where pre-training distillation might be very expensive and not even feasible. To investigate the usefulness of this layer selection technique and also how effective $AD^2$ is for models initialized with layer selection, we look into three selection strategies: **(1) Skip-layer selection strategy (*Skip*):** selects every other layer of the pre-trained teacher network, starting from the first layer of the network. **(2) Top-layer selection strategy (*Top*):** selects the top layers of the network. **(3) Bottom-layer selection strategy (*Bottom*):** selects the bottom layers of the network.

We apply the above selection strategies to a 6-layer DistillBERT to initialize a 3-layer student model and then fine-tune the student model on each downstream task. Table 4 shows that bottom-layer selection (*Bottom*) in general outperforms *Skip* and *Top* by 2.2 points (80.6 vs. 78.4) and 5.0 points (80.6 vs. 75.6) on average, respectively. *Top* leads to the worst accuracy, indicating that lower layers are the most important ones for adapting to downstream tasks. This is expected because bottom layers are closer to the input, which are more crucial for capturing the basic contextual information among tokens (Liu et al. 2019). Removing top layers yields the least accuracy drop. This is because top layers are biased towards the pre-training objective, which needs to be updated to adapt to downstream tasks anyway, as also observed by (Voita, Sennrich, and Titov 2019). Given these observations, we apply $AD^2$ to the best performing 3-layer model (*Bottom*) and observe that $AD^2$ provides consistent improvements in accuracy on all tested tasks. This result indicates that our approach is compatible and complementary with alternative student model initialization methods such as layer selection, which offers a solution to compress large-scale pre-trained Transformer models with flexibility and low cost (e.g., without pre-training distillation).

**Impact of an evolving teacher.** To evaluate how $AD^2$ performs as the teacher model evolves (e.g., by having larger sizes and stronger performance), we measure the difference between $BERT_{base}$ and $BERT_{large}$ teacher for model compression without and with $AD^2$. Results are summarized in Table 5. We observe that by simply changing the teacher model from $BERT_{base}$ to $BERT_{large}$, there is not much difference in student's performance when using just knowledge distillation. This observation is consistent with prior studies (Cho and Hariharan 2019; Sun et al. 2019), which shows that when the gap between the teacher and the student is large, it becomes more challenging for the student model to learn knowledge from the teacher. Interestingly, with adversarial data augmentation, we observe an overall 1.2 points improvement (85.7 vs. 84.5) when we have a stronger teacher, which indicates that $AD^2$ allows the student model to absorb more knowledge from stronger teachers. This is desirable because it allows the student model to adapt to the current state of the teacher model and supports a continuously evolving teacher that can better teach the student. More interestingly, we would like to highlight that $AD^2$ allows the $BERT_6$ student model to outperform $BERT_{12}$ on QQP, MRPC, and RTE, by distilling knowledge from $BERT_{24}$, indicating that our approach can also help a small model to achieve comparable or better performance than its larger counterpart when advised by an even stronger teacher.

| Model | MNLI-m -/mm (Acc.) | QQP (F1/Acc.) | QNLI (Acc.) | SST-2 (Acc.) | MRPC (F1/Acc.) | RTE (Acc.) | Avg. |
|---|---|---|---|---|---|---|---|
| Skip | 76.9/76.5 | 85.6/89.2 | 84.6 | 89.4 | 73.7/82.4 | 54.5 | 78.4 |
| Top | 71.4/71.6 | 83.2/87.7 | 77.3 | 86.1 | 72.0/81.4 | 55.5 | 75.6 |
| Bottom | 77.4/78.0 | 86.4/89.8 | 85.8 | 89.1 | 78.1/85.1 | 60.2 | 80.6 |
| + $AD^2$ | **79.4/79.8** | **87.2/90.5** | **87.4** | **90.4** | **81.3/87.4** | **62** | **82.3** |

Table 4: Comparison results of different layer selection strategies for initializing the student model. The results show that bottom-layer selection is more effective than other strategies, and $AD^2$ brings 1.7 points accuracy improvement to bottom layer selection based initialization.

| Teacher | MNLI | QQP | QNLI | SST-2 | MRPC | RTE | Avg. |
|---|---|---|---|---|---|---|---|
| $BERT_{12}$ + KD | 82.9/82.8 | 90.5 | 89 | 91.3 | 90.3 | 66 | 84.2 |
| $BERT_{24}$ + KD | 82.8/82.8 | 90.6 | 88.7 | 90.8 | 90.1 | 65.3 | 84.0 |
| $BERT_{24}$ + **$AD^2$** | **84.0/84.2** | **91.0** | **90.1** | **92.3** | **90.6** | **67.5** | **85.3** |

Table 5: Performance of the student model as the teacher evolves, without and with $AD^2$. The results show that while the student struggles to learn from a better teacher using existing knowledge distillation, $AD^2$ helps the student to achieve better accuracy as the teacher evolves.

## 5 Conclusion

In this work, we propose a lightweight and effective knowledge distillation approach, $AD^2$, for compressing pre-trained transformer models on low-resource downstream tasks. $AD^2$ leverages adversarial data augmentation in the distillation process, presenting more diverse data views to the student when transferring knowledge from the teacher model. Such a scheme prevents the student from overfitting on small domain-specific datasets, leading to improved generalization ability. Our empirical results suggest that $AD^2$ effectively and efficiently compresses pre-trained Transformers, improving the student model's accuracy. Our detailed analysis shows that this path has much potential for future work.

## References


Aguilar, G.; Ling, Y.; Zhang, Y.; Yao, B.; Fan, X.; and Guo, C. 2020. Knowledge Distillation from Internal Representations. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, 7350–7357. AAAI Press.

Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.; Srivastava, M. B.; and Chang, K. 2018. Generating Natural Language Adversarial Examples. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018*, 2890–2896. Association for Computational Linguistics.

Bommasani, R.; Hudson, D. A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M. S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. 2021. On the Opportunities and Risks of Foundation Models. *arXiv preprint arXiv:2108.07258*.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. *CoRR*, abs/2005.14165.

Bucila, C.; Caruana, R.; and Niculescu-Mizil, A. 2006. Model compression. In Eliassi-Rad, T.; Ungar, L. H.; Craven, M.; and Gunopulos, D., eds., *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*, 535–541. ACM.

Cai, Q.; Liu, C.; and Song, D. 2018. Curriculum Adversarial Training. In Lang, J., ed., *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 3740–3747. ijcai.org.

Cheng, Y.; Jiang, L.; and Macherey, W. 2019. Robust Neural Machine Translation with Doubly Adversarial Inputs. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 4324–4333. Association for Computational Linguistics.

Cho, J. H.; and Hariharan, B. 2019. On the Efficacy of Knowledge Distillation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, 4793–4801. IEEE.

Danskin, J. M. 2012. *The theory of max-min and its application to weapons allocation problems*, volume 5. Springer Science & Business Media.

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

Dong, L.; Yang, N.; Wang, W.; Wei, F.; Liu, X.; Wang, Y.; Gao, J.; Zhou, M.; and Hon, H.-W. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, 13042–13054.

Gan, Z.; Chen, Y.; Li, L.; Zhu, C.; Cheng, Y.; and Liu, J. 2020. Large-Scale Adversarial Training for Vision-and-Language Representation Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Gou, J.; Yu, B.; Maybank, S. J.; and Tao, D. 2021. Knowledge Distillation: A Survey. *Int. J. Comput. Vis.*, 129(6): 1789–1819.

Gupta, S.; Dube, P.; and Verma, A. 2020. Improving the affordability of robustness training for DNNs. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, 3383–3392. Computer Vision Foundation / IEEE.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *CoRR*, abs/1503.02531.

Jia, R.; and Liang, P. 2017. Adversarial Examples for Evaluating Reading Comprehension Systems. In Palmer, M.; Hwa, R.; and Riedel, S., eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, 2021–2031. Association for Computational Linguistics.

Jiang, H.; He, P.; Chen, W.; Liu, X.; Gao, J.; and Zhao, T. 2020. SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2177–2190. Association for Computational Linguistics.

Jiao, X.; Yin, Y.; Shang, L.; Jiang, X.; Chen, X.; Li, L.; Wang, F.; and Liu, Q. 2019. TinyBERT: Distilling BERT for Natural Language Understanding. *CoRR*, abs/1909.10351.

Jin, C.; Netrapalli, P.; and Jordan, M. 2020. What is local optimality in nonconvex-nonconcave minimax optimization? In *International Conference on Machine Learning*, 4880–4889. PMLR.

Kaplan, J.; McCandlish, S.; Henighan, T.; Brown, T. B.; Chess, B.; Child, R.; Gray, S.; Radford, A.; Wu, J.; and Amodei, D. 2020. Scaling Laws for Neural Language Models. *CoRR*, abs/2001.08361.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kobayashi, S. 2018. Contextual Augmentation: Data Augmentation by Words with Paradigmatic Relations. In Walker, M. A.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, 452–457. Association for Computational Linguistics.

Liu, N. F.; Gardner, M.; Belinkov, Y.; Peters, M. E.; and Smith, N. A. 2019. Linguistic Knowledge and Transferability of Contextual Representations. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 1073–1094. Association for Computational Linguistics.

Liu, X.; Cheng, H.; He, P.; Chen, W.; Wang, Y.; Poon, H.; and Gao, J. 2020. Adversarial Training for Large Neural Language Models. *CoRR*, abs/2004.08994.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Rivera, A.; Wang, H.; and Xu, H. 2018. The Online Saddle Point Problem and Online Convex Optimization with Knapsacks. *arXiv preprint arXiv:1806.08301*.

Sajjad, H.; Dalvi, F.; Durrani, N.; and Nakov, P. 2020. Poor Man's BERT: Smaller and Faster Transformer Models. *CoRR*, abs/2004.03844.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Shalev-Shwartz, S.; et al. 2011. Online learning and online convex optimization. *Foundations and trends in Machine Learning*, 4(2): 107–194.

Sun, S.; Cheng, Y.; Gan, Z.; and Liu, J. 2019. Patient Knowledge Distillation for BERT Model Compression. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 4322–4331. Association for Computational Linguistics.

Sun, Z.; Yu, H.; Song, X.; Liu, R.; Yang, Y.; and Zhou, D. 2020. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2158–2170. Association for Computational Linguistics.

Tang, R.; Lu, Y.; Liu, L.; Mou, L.; Vechtomova, O.; and Lin, J. 2019. Distilling Task-Specific Knowledge from BERT into Simple Neural Networks. *CoRR*, abs/1903.12136.

Turc, I.; Chang, M.; Lee, K.; and Toutanova, K. 2019. Well-Read Students Learn Better: The Impact of Student Initialization on Knowledge Distillation. *CoRR*, abs/1908.08962.

Voita, E.; Sennrich, R.; and Titov, I. 2019. The Bottom-up Evolution of Representations in the Transformer: A Study with Machine Translation and Language Modeling Objectives. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 4395–4405. Association for Computational Linguistics.

Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *7th International Conference on Learning Representations*.

Wang, D.; Gong, C.; and Liu, Q. 2019. Improving Neural Language Modeling via Adversarial Training. In Chaudhuri, K.; and Salakhutdinov, R., 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*, 6555–6565. PMLR.

Wang, L.; and Yoon, K. 2020. Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks. *CoRR*, abs/2004.05937.

Wang, W.; Bao, H.; Huang, S.; Dong, L.; and Wei, F. 2020a. MiniLMv2: Multi-Head Self-Attention Relation Distillation for Compressing Pretrained Transformers. *CoRR*, abs/2012.15828.

Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; and Zhou, M. 2020b. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Wei, J. W.; and Zou, K. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 6381–6387. Association for Computational Linguistics.

Wu, X.; Lv, S.; Zang, L.; Han, J.; and Hu, S. 2019. Conditional BERT Contextual Augmentation. In Rodrigues, J. M. F.; Cardoso, P. J. S.; Monteiro, J. M.; Lam, R.; Krzhizhanovskaya, V. V.; Lees, M. H.; Dongarra, J. J.; and Sloot, P. M. A., eds., *Computational Science - ICCS 2019 - 19th International Conference, Faro, Portugal, June 12-14, 2019, Proceedings, Part IV*, volume 11539 of *Lecture Notes in Computer Science*, 84–95. Springer.

Xie, C.; Tan, M.; Gong, B.; Wang, J.; Yuille, A. L.; and Le, Q. V. 2020. Adversarial Examples Improve Image Recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, 816–825. Computer Vision Foundation / IEEE.

Ye, N.; Li, Q.; Zhou, X.; and Zhu, Z. 2020. Amata: An Annealing Mechanism for Adversarial Training Acceleration. *CoRR*, abs/2012.08112.

Zhao, S.; Gupta, R.; Song, Y.; and Zhou, D. 2021. Extremely Small BERT Models from Mixed-Vocabulary Training. In Merlo, P.; Tiedemann, J.; and Tsarfaty, R., eds., *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*, 2753–2759. Association for Computational Linguistics.

Zhu, C.; Cheng, Y.; Gan, Z.; Sun, S.; Goldstein, T.; and Liu, J. 2020. FreeLB: Enhanced Adversarial Training for Natural Language Understanding. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zinkevich, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th international conference on machine learning (icml-03)*, 928–936.