AD-KD: Attribution-Driven Knowledge Distillation for Language Model Compression

Siyue Wu[1], Hongzhan Chen[1], Xiaojun Quan[1]*, Qifan Wang[2] and Rui Wang[3]

[1]School of Computer Science and Engineering, Sun Yat-sen University, China

[2]Meta AI

[3]Vipshop (China) Co., Ltd., China

[1] {wusy39, chenhzh59}@mail2.sysu.edu.cn, quanxj3@mail.sysu.edu.cn

[2]wqfcr@fb.com

[3]mars198356@hotmail.com

Abstract

Knowledge distillation has attracted a great deal of interest recently to compress pre-trained language models. However, existing knowledge distillation methods suffer from two limitations. First, the student model simply imitates the teacher's behavior while ignoring the underlying reasoning. Second, these methods usually focus on the transfer of sophisticated model-specific knowledge but overlook data-specific knowledge. In this paper, we present a novel attribution-driven knowledge distillation approach, which explores the token-level rationale behind the teacher model based on Integrated Gradients (IG) and transfers attribution knowledge to the student model. To enhance the knowledge transfer of model reasoning and generalization, we further explore multi-view attribution distillation on all potential decisions of the teacher. Comprehensive experiments are conducted with BERT on the GLUE benchmark. The experimental results demonstrate the superior performance of our approach to several state-of-the-art methods.

1 Introduction

Transformer-based pre-trained language models (PLMs), such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), have aroused widespread interest among Natural Language Processing (NLP) researchers in recent years. These language models are first pre-trained on large-scale unlabeled corpora to learn the general representation of language, and then fine-tuned on specific downstream tasks to effectively transfer the general knowledge to target domains. This pre-training and fine-tuning paradigm leads to state-of-the-art performances in various NLP tasks such as natural language understanding. However, with the rapid growth of the model scale, the deployment of large-scale PLMs becomes challenging, especially in low-resource scenarios. To this end, a variety of model compression techniques have been developed. Among them, knowledge distillation (KD) (Hinton et al., 2015) is a newly emerging technology that aims to obtain a small student model by distilling knowledge from a large teacher model and achieve comparable performance.

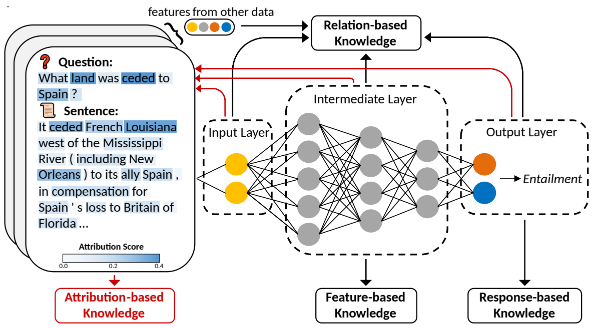

Figure 1: An example from the QNLI dataset (Rajpurkar et al., 2016) to illustrate different knowledge distillation techniques including the proposed attribution-driven method. Darker colors mean larger attribution scores.

Existing knowledge distillation methods can be divided into three categories, namely response-based, feature-based, and relation-based (Gou et al., 2021). While response-based methods (Turc et al., 2019) directly distill the final output, e.g. probability distribution, from the top of the teacher, feature-based (Sun et al., 2019) and relation-based methods (Liu et al., 2022) try to align the features from intermediate layers of teacher and student models and minimize the difference. To transfer comprehensive knowledge from the teacher, a common practice is to combine response-based methods with the other two (Park et al., 2021). However, due to the capacity gap between the teacher and the student, feature-based and relation-based methods may not necessarily bring improvement to response-based methods (Liang et al., 2022). To sum up, existing knowledge distillation methods have two limitations. First, they mainly focus on understanding what the teacher's behavior is, instead of why the teacher behaves like this, hinder-

*Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics Volume 1: Long Papers, pages 8449–8465

ing the reasoning and generalization ability of the student model. Second, they pay more attention to distilling sophisticated model-specific knowledge from intermediate layers but neglect data-specific knowledge, which may contain valuable rationale information to understand how the teacher model arrives at a prediction.

To address the above limitations, in this paper we propose a novel Attribution-Driven Knowledge Distillation (AD-KD) approach that transfers attribution-based knowledge from the teacher to the student. As shown in Figure 1, the attribution information reflects the importance of different tokens towards the prediction, which contains reasoning knowledge of the model and can be complementary to the soft-label knowledge. By transferring such attribution knowledge, the student is allowed to learn the token-level rationale behind the teacher's behavior and thus generalizes better. Specifically, we utilize Integrated Gradients (IG) (Sundararajan et al., 2017), a well-established gradient-based attribution method, to calculate the importance score of each input token. To reduce the influence of trivial dimensions in the teacher's input embeddings, we further adopt the top-K strategy to filter out dimensions with low attribution scores. The remaining attribution scores are aggregated and normalized to denote the importance of individual tokens. Moreover, we extract the attribution knowledge for all possible predictions rather than just the prediction with the highest probability. By transferring the multi-view attribution knowledge, the student learns a more comprehensive understanding of the teacher's soft-label distribution.

Extensive experiments are conducted with BERT (Devlin et al., 2019) on the GLUE benchmark (Wang et al., 2018). The experimental results demonstrate the effectiveness and superiority of our approach over several state-of-the-art baselines. Furthermore, we show that attribution knowledge from different layers contains different information, while the input layer contains the most prominent attribution knowledge for distillation. To summarize, the main contributions are threefold. First, we propose a novel attribution-driven knowledge distillation framework for language model compression that effectively transfers attribution knowledge from the teacher to the student. Second, we extract multi-view attribution knowledge based on model predictions to learn comprehensive reasoning knowledge. Third, we systematically validate AD-KD on the GLUE benchmark and show its superior performance over state-of-the-art baselines.

2 Related Work

2.1 Knowledge Distillation

Knowledge distillation methods can be divided into three categories, namely response-based, feature-based and relation-based KD (Gou et al., 2021). Response-based KD was first proposed by Hinton et al. (2015), where the final output is adopted to transfer the label knowledge. Sanh et al. (2019) and Turc et al. (2019) applied this idea to BERT and yielded smaller models with minor performance drops. Recently, feature-based and relation-based distillation methods have drawn a lot of attention, which transfer knowledge contained in the intermediate layers to the student. For feature-based methods, Sun et al. (2019) first regarded the hidden representations of the [CLS] token as hints to extract sentence-level features from the teacher. Jiao et al. (2020) and Sun et al. (2020b) further matched the hidden representations of all tokens between teacher and student models. Sun et al. (2020a) proposed contrastive distillation on intermediate representations. As for relation-based methods, Park et al. (2021) proposed CKD which adopts pair-wise distance and triple-wise angle to model the sophisticated relations among token representations from both horizontal and vertical directions. Based on CKD, Liu et al. (2022) further extracted structural relations from multi-granularity representations and distilled this kind of well-organized multi-granularity structural knowledge hierarchically across layers. Wang et al. (2020, 2021) generalized the conventional query-key attention to query-query attention, key-key attention, and value-value attention. Different from these methods, we investigate knowledge distillation from the attribution perspective, which reveals the teacher's reasoning behavior and can be used to transfer comprehensive data-specific knowledge. More details about the differences between existing methods and ours are discussed in Appendix B.

2.2 Attribution

Attribution analysis (Baehrens et al., 2010; Ancona et al., 2018) aims at assigning importance scores to intermediate or input features of a network. Occlusion-based methods (Zeiler and Fergus, 2014) compute the importance score of each feature by erasing that feature and measuring the difference between new output and the original output. However, occlusion-based methods need to forward pass the model once for each feature, leading to low computational efficiency. To address this issue, gradient-based methods (Li et al., 2016; Ding et al., 2019; Brunner et al., 2020; Sundararajan et al., 2017) exploit the gradient information of features to approximate occlusion-based methods, which only require a single forward process. Similarly, propagation-based methods (Bach et al., 2015; Shrikumar et al., 2017) modify the back-propagation rules to redistribute the model output among the target features along the back-propagation path. Perturbation-based methods (Guan et al., 2019; Schulz et al., 2020; De Cao et al., 2020) add noise to features to examine their importance for model predictions. Attribution has been adopted in model compression techniques such as pruning (Michel et al., 2019) and adaptive inference (Modarressi et al., 2022) but has not been explored in knowledge distillation. In this work, we take the initiative to investigate the effect of attribution in knowledge distillation.

3 Methodology

3.1 Preliminary

Integrated Gradients (Sundararajan et al., 2017) is a theoretically tenable method to attribute the prediction of a deep network to its input or intermediate features. Formally, given a feature $\mathbf{x} = [x_1, x_2, ..., x_n] \in \mathbb{R}^n$ with a baseline feature $\mathbf{x}' = [x'_1, x'_2, ..., x'_n] \in \mathbb{R}^n$, and the model function $F(.)$, IG leverages integral to represent the difference between $F(\mathbf{x})$ and $F(\mathbf{x}')$ by selecting a straight line path from $\mathbf{x}'$ to $\mathbf{x}$ as the integral path:

$$F(\mathbf{x}) - F(\mathbf{x}') = \sum_{i=1}^{n} \mathrm{IG}_i(F, \mathbf{x}) = \sum_{i=1}^{n} [(x_i - x'_i) \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha]. \quad (1)$$

In practice, continual integral can be approximated by discrete summation:

$$\mathrm{IG}_i^{approx}(F, \mathbf{x}) = (x_i - x'_i) \times \sum_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i} \times \frac{1}{m}, \quad (2)$$

where m is the number of summation steps (a bigger m usually results in better approximation). Intuitively, the magnitude of integrated gradient indicates its importance while its sign illustrates the positive or negative effect on the target output.

In this paper, we focus on Transformer-based architecture and attribute the model prediction to input features. With slight abuse of notation, we denote the input sequence as $\mathbf{x} = [x_1, x_2, ..., x_n]$, where n is the sequence length and each x_i represents a token. Transformer first converts the token sequence to d-dimensional embedding sequence $\mathbf{E} = [\mathbf{e_1}, \mathbf{e_2}, ..., \mathbf{e_n}] \in \mathbb{R}^{n \times d}$ through the embedding layer. And then the contextualized representations $\mathbf{H} = \mathrm{Transformer}(\mathbf{E}) \in \mathbb{R}^{n \times d}$ are obtained after several layers of Transformer blocks. Finally, a task-specific head is applied on $\mathbf{H}$ to get the final output $P = [P_1, P_2, ..., P_C] \in \mathbb{R}^C$, which is typically a probability distribution. Denote the mapping function $\mathbf{E} \to P_c$ as $F^c(.)$, where c represents the label of interest. In this case, our attribution map is computed on each individual dimension of each input embedding, which is denoted as e_{ij}:

$$\mathrm{IG}_{ij}^{approx}(F^c, \mathbf{E}) = (e_{ij} - e'_{ij}) \times \sum_{k=1}^{m} \frac{\partial F^c(\mathbf{E}' + \frac{k}{m} \times (\mathbf{E} - \mathbf{E}'))}{\partial e_{ij}} \times \frac{1}{m}. \quad (3)$$

In the implementation, we stack n [PAD] token embeddings as baseline features $\mathbf{E}'$ since they usually have no influence on the model prediction.

3.2 AD-KD

In this section, we elaborate on our proposed Attribution-Driven Knowledge Distillation (AD-KD), including attribution maps and attribution distillation. The overall framework of AD-KD is illustrated in Figure 2.

3.2.1 Attribution Maps

The attribution scores of a language model reflect the importance of different tokens towards the prediction, which contains valuable data-specific reasoning knowledge. The scores are computed among different tokens at different dimensions of a given model, using IG defined in Section 3.1. In this work, we do not take the sign into consideration, since the scores at different dimensions of the same token embedding would cannibalize each other when combining them into a token-level attribution score. This observation is consistent with the findings in (Atanasova et al., 2020).

When calculating the attribution scores, we observed that there exist certain dimensions whose attribution scores remain relatively low across different tokens. The attribution scores from these dimensions minimize the difference between important and unimportant tokens, which can be regarded

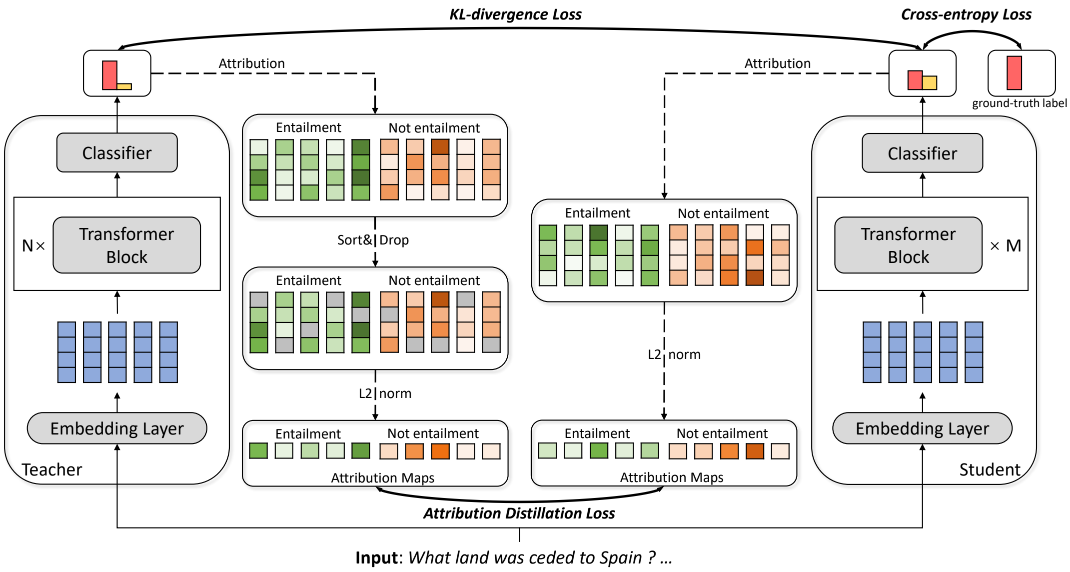

Figure 2: Overview of our AD-KD framework. The example in Figure 1 is taken as the input. AD-KD first extracts the attribution maps from the teacher model and then transfers the attribution-based knowledge to the student.

as noises. For better illustration, Figure 3 shows an example of sentence "seem weird and distanced" whose annotation is *negative* sentiment. It is clear that "weird" and "distance" are the keywords that contribute most to the prediction, whereas a proportion of dimensions of them present low attribution scores. To alleviate the influence of noisy dimensions in the input embeddings, we simply choose the top-K dimensions with high attribution scores and filter out dimensions with low attribution scores. Formally, the attribution score of token x_i with respect to the label c in the teacher model can be calculated as:

$$a_i^{t,c} = \|\text{TopK}(\text{IG}_i^{approx}(F^{t,c}, \mathbf{E}^t))\|_2, \tag{4}$$

where the superscript t denotes the teacher model. Therefore, the attribution map of the teacher consists of a sequence of attribution scores:

$$\mathbf{a}^{t,c} = [a_1^{t,c}, a_2^{t,c}, ..., a_n^{t,c}]. \tag{5}$$

For the student, the extraction of attribution map is similar except that we consider all dimensions for two reasons. First, it reduces the difficulty of training. Second, the student is allowed to learn from the noiseless attribution map of the teacher.

$$\begin{aligned} a_i^{s,c} &= \|\text{IG}_i^{approx}(F^{s,c}, \mathbf{E}^s)\|_2, \\ \mathbf{a}^{s,c} &= [a_1^{s,c}, a_2^{s,c}, ..., a_n^{s,c}]. \end{aligned} \tag{6}$$

Considering that the teacher can make multiple decisions, each of which is associated with a probability, we further propose to extract multi-view attribution knowledge. Specifically, we extract the attribution maps for all possible predictions of the model rather than a single prediction, e.g., the prediction with the maximum probability or the prediction corresponding to the ground-truth label. By transferring the multi-view attribution knowledge, the student can capture a more comprehensive understanding of the teacher's soft-label distribution. The multi-view attribution maps are defined as:

$$\mathbf{A}^t = \|_{c=1}^C \, \mathbf{a}^{t,c}, \; \mathbf{A}^s = \|_{c=1}^C \, \mathbf{a}^{s,c}, \tag{7}$$

where $\|$ is the concatenation operation.

3.2.2 Attribution Distillation

Given the multi-view attribution maps, a straightforward strategy to transfer the knowledge is to directly minimize the difference between the two sets of maps in teacher and student models, with distance metrics like L2 distance (MSE):

$$\|\mathbf{A}^t - \mathbf{A}^s\|_2. \tag{8}$$

However, one obvious shortcoming with this approach is that there may exist a magnitude gap between the attribution scores in teacher and student models at the early phase of distillation, since the teacher is already well-trained while the student has little attribution knowledge. Under this circumstance, the student is likely to fall into a local optimum. To enable smooth knowledge distillation, we normalize the attribution maps before minimizing the difference. Concretely, we first transform the single-view attribution maps into unit vectors:

$$\widetilde{\mathbf{a}}^{t,c} = \frac{\mathbf{a}^{t,c}}{\|\mathbf{a}^{t,c}\|_2}, \; \widetilde{\mathbf{a}}^{s,c} = \frac{\mathbf{a}^{s,c}}{\|\mathbf{a}^{s,c}\|_2}. \tag{9}$$

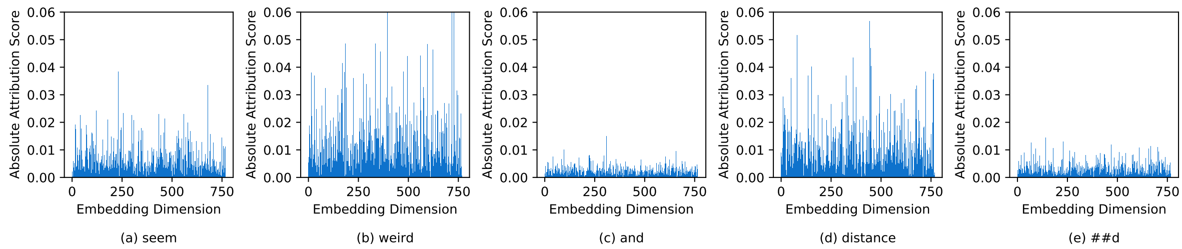

Figure 3: An example from the SST-2 dataset (Socher et al., 2013). Given the sentence "seem weird and distanced" and its sentiment label *negative*, the distributions of absolute attribution scores among different tokens and dimensions are shown in subfigures (a)-(e). The model is a well-trained BERT$_{base}$ (teacher) and the IG steps m is set to 1.

Then we reformulate the normalized multi-view attribution maps in Eq. (7) as:

$$\widetilde{\mathbf{A}}^t = \|_{c=1}^C \widetilde{\mathbf{a}}^{t,c}, \ \widetilde{\mathbf{A}}^s = \|_{c=1}^C \widetilde{\mathbf{a}}^{s,c}. \tag{10}$$

The normalized attribution maps only preserve the information of relative importance among tokens regardless of their absolute importance, which we believe is the crucial knowledge to transfer. Finally, we define the attribution distillation loss as:

$$\mathcal{L}_{attr} = \|\widetilde{\mathbf{A}}^t - \widetilde{\mathbf{A}}^s\|_2. \tag{11}$$

3.2.3 Overall Objective

We combine the original cross-entropy loss between the output of the student and the ground-truth label, the response-based loss (on the logits) (Hinton et al., 2015), and the proposed attribution-driven distillation loss to train the student model. The overall objective is defined as:

$$\mathcal{L} = (1-\alpha)\mathcal{L}_{ce} + \alpha\mathcal{L}_{logit} + \beta\mathcal{L}_{attr}, \tag{12}$$

where $\mathcal{L}_{ce} = -\log\sigma(z^s)[y]$ is the cross-entropy loss and $\mathcal{L}_{logit}$=KL$(\sigma(\frac{z^t}{\tau})\|\sigma(\frac{z^s}{\tau}))$ is the loss on the output logits. And, α and β are two hyperparameters, σ is the softmax function, y is the ground-truth label, τ is the temperature, and z^t and z^s are the output logits of the teacher and student models, respectively. KL$(\cdot)$ denotes the KL-divergence.

4 Experimental Settings

4.1 Datasets

We evaluate our method on eight tasks of the GLUE benchmark (Wang et al., 2018), including CoLA (Warstadt et al., 2019), MNLI (Williams et al., 2018), SST-2 (Socher et al., 2013), QNLI (Rajpurkar et al., 2016), MRPC (Dolan and Brockett, 2005), QQP (Chen et al., 2018), RTE (Bentivogli et al., 2009) and STS-B (Cer et al., 2017). The details of these datasets are introduced in Appendix A.1. For evaluation metrics, we follow previous works (Park et al., 2021; Liu et al., 2022) and report accuracy on MNLI, SST-2, QNLI, QQP and RTE, F1 score on MRPC, Matthews correlation coefficient on CoLA, and Spearman's rank correlation coefficient on STS-B.

4.2 Baseline Methods

We compare AD-KD with response-based KD methods and several state-of-the-art feature-based and relation-based KD methods. Response-based baselines include Vanilla KD (Hinton et al., 2015) and PD (Turc et al., 2019). Feature-based and relation-based baselines include PKD (Sun et al., 2019) which distills the hidden representations, TinyBERT (Jiao et al., 2020) which distills the self-attention matrices, and CKD (Park et al., 2021) and MGSKD (Liu et al., 2022) which distill the relation between hidden representations. For a fair comparison, MiniLM (Wang et al., 2020, 2021) and MobileBERT (Sun et al., 2020b) are not presented due to their two-stage distillation settings which involve both task-agnostic and task-specific distillation. Our AD-KD focuses on task-specific distillation and does not augment the training sets. Moreover, MGSKD (Liu et al., 2022) only reports results on a 4-layer BERT student model which is different from other baselines. To ensure a fair comparison, we re-implemented MGSKD using their released code to obtain a 6-layer student model. The original MGSKD approach also relies on span-level information that is extracted from external knowledge sources, which is not publicly available nor included in other baselines. Therefore, we did not use this external knowledge in our re-implementation of MGSKD.

4.3 Implementation Details

Our code is implemented in Pytorch with the Transformers package (Wolf et al., 2020). We fine-

Model	#Params	CoLA (Mcc)	MNLI-(m/mm) (Acc)	SST-2 (Acc)	QNLI (Acc)	MRPC (F1)	QQP (Acc)	RTE (Acc)	STS-B (Spear)	Avg
Dev										
$BERT_{base}$ (Teacher)	110M	60.3	84.9/84.8	93.7	91.7	91.4	91.5	69.7	89.4	84.1
$BERT_6$ (Student)	66M	51.2	81.7/82.6	91.0	89.3	89.2	90.4	66.1	88.3	80.9
Vanilla KD (Hinton et al., 2015)	66M	53.6	82.7/83.1	91.1	90.1	89.4	90.5	66.8	88.7	81.6
PD (Turc et al., 2019)	66M	-	82.5/83.4	91.1	89.4	89.4	90.7	66.7	-	-
PKD (Sun et al., 2019)	66M	45.5	81.3/-	91.3	88.4	85.7	88.4	66.5	86.2	79.2
TinyBERT (Jiao et al., 2020)	66M	53.8	83.1/83.4	92.3	89.9	88.8	90.5	66.9	88.3	81.7
CKD (Park et al., 2021)	66M	55.1	**83.6**/ 84.1	**93.0**	90.5	89.6	**91.2**	67.3	89.0	82.4
MGSKD (Liu et al., 2022)	66M	49.1	83.3/83.9	91.7	90.3	89.8	**91.2**	67.9	88.5	81.5
AD-KD	66M	**58.3**	83.4/**84.2**	91.9	**91.2**	**91.2**	**91.2**	**70.9**	**89.2**	**83.4**
Test										
$BERT_{base}$ (Teacher)	110M	51.5	84.5/84.1	94.1	90.9	87.7	89.2	67.5	85.5	81.4
$BERT_6$ (Student)	66M	41.7	81.9/81.0	91.3	88.9	85.2	88.0	64.0	82.4	77.9
Vanilla KD (Hinton et al., 2015)	66M	42.3	82.7/81.8	92.0	89.3	86.3	88.2	65.0	82.7	78.6
PD (Turc et al., 2019)	66M	-	82.8/82.2	91.8	88.9	86.8	88.9	65.3	-	-
PKD (Sun et al., 2019)	66M	43.5	81.5/81.0	92.0	89.0	85.0	88.9	65.5	81.6	78.4
MGSKD (Liu et al., 2022)	66M	42.8	**83.4/82.8**	**92.1**	89.5	87.0	**89.1**	63.7	82.2	78.7
AD-KD	66M	**47.0**	83.1/82.6	91.8	**90.0**	**87.1**	88.9	**65.8**	**83.4**	**79.6**

Table 1: Overall results on the GLUE benchmark. The results of baselines except vanilla KD and MGSKD are imported from Park et al. (2021). Results of development sets are averaged over 3 runs and we submit the model with the highest score to the official GLUE server to obtain the results of test sets. Average score is computed excluding the MNLI-mm accuracy. The best results of the student models are shown in bold and the second best results are shown with underline. Results are statistically significant with p-value < 0.005.

tune $BERT_{base}$ as the teacher model, and utilize a smaller BERT released by Turc et al. (2019) with 6 Transformer layers, 768 hidden neurons and 12 attention heads to instantiate the student model following Park et al. (2021). We search for the optimal learning rate in {2e-5, 3e-5, 4e-5, 5e-5}, α in {0.8, 0.9, 1.0} and temperature τ in {1, 2, 3, 4}. For the hyperparameter β, we tune within {1, 10, 50, 100}. For the IG steps m described in Section 3.1, we adopt m = 1 in the main results due to the huge computational overhead. Part of results with m varying from 1 to 8 are reported in Section 5.4. K is empirically searched within {384, 512, 640, 700, 734, 768}. Results with different values of K are also reported. The detailed hyperparameter settings and training cost are provided in Appendix A.2. Our code is available at https://github.com/brucewsy/AD-KD.

5 Results and Analysis

5.1 Main Results

The main results are presented in Table 1. It can be seen that AD-KD outperforms all baselines on most of the datasets. Specifically, AD-KD yields an average improvement of 1.0 and 1.9 points over CKD and MGSKD respectively on development sets, and another average improvement of 0.9 points over MGSKD on test sets. Note that other feature-based and relation-based KD methods even underperform vanilla KD, indicating the difficulty of aligning the teacher and the student at intermediate layers. In contrast, AD-KD distills the attribution knowledge from a global perspective which is more data-specific and shows significant improvement over vanilla KD. We provide two cases in Appendix C.3 to intuitively demonstrate the strength of AD-KD. We also observe that AD-KD does not show a satisfying performance on SST-2. We believe the reason is that the sentences in SST-2 are much shorter than those in other datasets, and in this case, the student is likely to already capture the attribution knowledge implicitly from the soft-labels of the teacher (Zhang et al., 2022).

5.2 Ablation Study

Impact of Loss Terms To analyze the impact of different loss terms, we conduct ablation experiments on three variants of AD-KD: (1) AD-KD without attribution distillation (i.e., vanilla KD), (2) AD-KD without the original cross-entropy loss, and (3) AD-KD without logit distillation. As reported in Table 2, again we observe an obvious performance drop after removing the attribution distillation. We also note that removing either the conventional cross-entropy loss or logit distillation loss causes noticeable performance degradation, suggesting both of them contribute to the improve-

Method	CoLA (Mcc)	MNLI-(m/mm) (Acc)	SST-2 (Acc)	QNLI (Acc)	MRPC (F1)	QQP (Acc)	RTE (Acc)	STS-B (Spear)
AD-KD	58.3	83.4/84.2	91.9	91.2	91.2	91.2	70.9	89.2
w/o $\mathcal{L}_{attr}$	53.6	82.7/83.1	91.2	90.2	89.2	90.5	67.5	88.9
w/o $\mathcal{L}_{ce}$	57.8	83.6/84.1	91.3	90.8	90.8	91.2	69.3	88.9
w/o $\mathcal{L}_{logit}$	53.9	81.9/82.8	91.1	90.5	89.9	90.9	68.6	88.8

Table 2: Ablation study of different loss terms. The results are based on GLUE development sets.

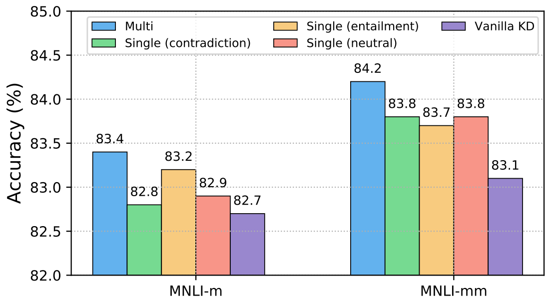

Figure 4: Ablation study of multi-view attribution on the MNLI development set.

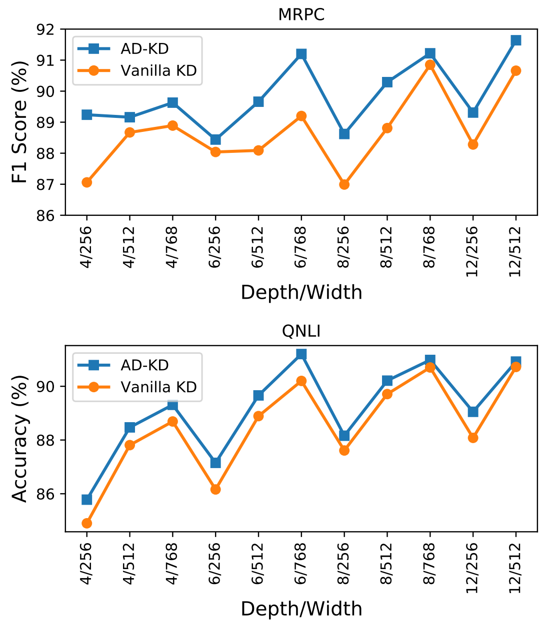

Figure 5: Results of AD-KD and vanilla KD on MRPC and QNLI development sets at different student scales.

ment of AD-KD. Nevertheless, our attribution distillation contributes most to the performance of AD-KD, showing that data-specific reasoning information is crucial in knowledge distillation.

Multi-view Attribution In AD-KD, the student learns the attribution knowledge from a variety of possible outputs to get a better understanding of the teacher. Here we study how the number of attribution views affects the final results. Experiments are conducted on MNLI which is a multi-classification task including three labels: entailment, contradiction, and neutral. We make a comparison between multi-view attribution and single-view attribution w.r.t. each candidate label respectively. The results are shown in Figure 4, from which we note that each of the single-view attributions plays a positive role and is superior to vanilla KD. Moreover, combining all attribution views yields further performance improvement, demonstrating that multi-view attribution is more preferable for distillation.

Student Model Size To investigate whether AD-KD can boost the performance across different sizes of student, we further compare AD-KD with vanilla KD on MRPC and QNLI under various student scales provided by Turc et al. (2019). As observed in Figure 5, AD-KD consistently outperforms vanilla KD, which validates the effectiveness and stability of our approach.

5.3 Impact of Top-K

Recall that in order to eliminate the interference of noisy dimension, AD-KD adopts the top-K approach on the input embeddings of the teacher to filter out the dimensions with relatively low attribution scores. In this section, we conduct in-depth analysis on the impact of K. We conduct experiments on STS-B and QNLI, and plot the results with different values of K in Figure 6. As illustrated in the figure, the performance on the small dataset STS-B (7k) first improves as K increases and then slightly degrades after K exceeds 600. However, the performance on the larger dataset QNLI (108k) improves almost monotonically with the increasing of K. We conjecture that choosing a suitable K is beneficial on small datasets since there are probably more noisy dimensions in the input embeddings of the teacher, while preserving all dimensions may be preferable on larger datasets.

5.4 Impact of IG Steps

In our experiments, the IG steps m are set to 1 by default when extracting the attribution maps. In this section, we provide more results with different values of m in Figure 7 to understand its impact on distillation. We observe that as m increases,

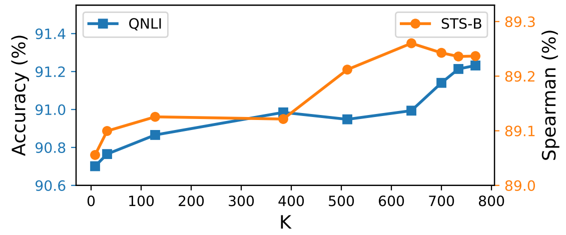

Figure 6: Results on STS-B and QNLI development sets as the number (K) of retained dimensions changes.

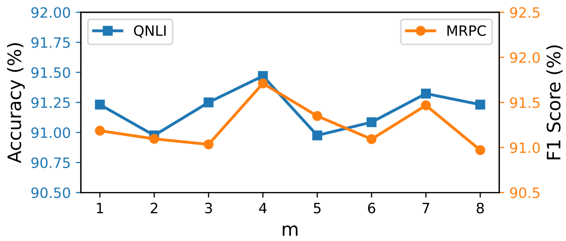

Figure 7: Results on MRPC and QNLI development sets as the number (m) of IG steps changes.

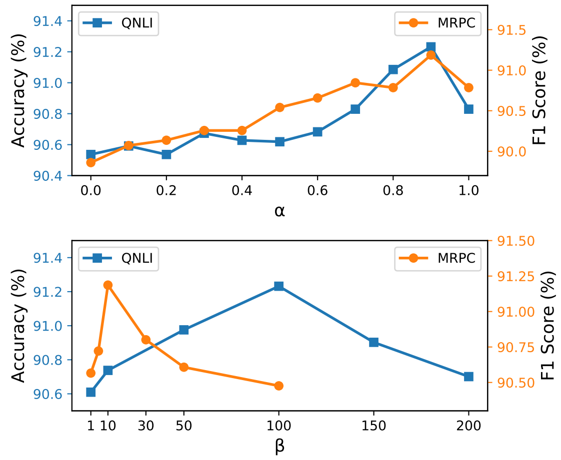

Figure 8: Results on MRPC and QNLI development sets as α and β changes.

the performance of AD-KD fluctuates in a certain range. Although it is possible to find a point that surpasses our default setting and even the teacher, identifying the optimal value of m for each task is costly since a large m causes huge computational overhead. In contrast, m=1 achieves better trade-off between performance and computational cost.

Attribution Layer	MRPC (F1)	QNLI (Acc)
input	**91.2**	**91.2**
first	90.5	90.9
penultimate	90.4	90.9
uniform	90.6	91.1
input & uniform	90.1	90.6

Table 3: Results of different attribution layers on MRPC and QNLI development sets.

5.5 Attribution Distillation Layer

Apart from the attribution knowledge of input layer, the attribution knowledge of intermediate layers can also be transferred during distillation. To confirm the motivation that the former is better than the latter, we conduct experiments on MRPC and QNLI with different attribution layers. Specifically, we choose the first layer and the penultimate layer for comparison. Besides, we also try a uniform strategy which is widely adopted as the mapping function between the teacher and the student layers (Jiao et al., 2020; Park et al., 2021; Liu et al., 2022). From the results shown in Table 3, we see that uniform mapping strategy performs best among intermediate layer methods. However, neither of these intermediate layers outperforms input layer, indicating that the attribution knowledge of intermediate layers is more model-specific and difficult to transfer. In addition, distilling the knowledge jointly from the input and the intermediate layers does not improve the performance.

5.6 Impact of α and β

For the training objective of AD-KD, we introduce α and β to balance the original cross-entropy loss, logit distillation loss, and attribution distillation loss. To investigate their impact on model performance, we show the results of different values of α and β on MRPC and QNLI in Figure 8, where we fix one while altering the other. We observe a unified trend across different tasks that when α is small, the student does not perform well due to the lack of response-based knowledge of the teacher, and when α is around 0.9, the student performs best. Therefore, we select α close to 1. We also observe from the figure that as β increases, the performance first keeps improving and reaches the peak, then it starts to decline. Unlike α, however, the optimal value of β varies with different tasks, indicating that β is more sensitive to the task compared to α. More discussion of β are given in Appendix C.2.

6 Conclusion

In this paper, we propose AD-KD, a novel knowledge distillation framework for language model compression. Unlike other distillation methods, AD-KD investigates the model knowledge from the perspective of input attribution, which is vital yet easy to transfer between the teacher and the student. Moreover, top-K method is adopted to obtain noiseless attribution maps among input tokens, and multi-view attribution is conducted for a more comprehensive distillation. To our knowledge, this is the first work that incorporates attribution into knowledge distillation. Extensive experiments including ablation studies are carried out to show the effectiveness of AD-KD and its components. With the recent emergence of large language models (LLMs), gradient-based attribution methods are infeasible due to the unavailable parameters. However, the idea of AD-KD can still be potentially extended to these black-box models by using occlusion-based attribution or using chain-of-thoughts (Wei et al., 2022) as the rationale for distillation. We will leave it to future work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 62176270), the Guangdong Basic and Applied Basic Research Foundation (No. 2023A1515012832), and the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No. 2017ZT07X355).

Limitations

This work introduces the general idea of incorporating attribution into knowledge distillation, and there are three potential limitations. First, although AD-KD chooses Integrated Gradients for attribution, there are actually other attribution methods (Janizek et al., 2021; Sikdar et al., 2021) which can also be fitted in our framework. The question of whether these methods perform better than Integrated Gradients when combined with knowledge distillation is still unclear. Second, we conduct experiments on BERT of different scales and have not yet validated the effectiveness of AD-KD on other model structures. Third, while we only perform task-specific knowledge distillation in our experiments, applying AD-KD to task-agnostic knowledge distillation is also worth investigating.

Ethics Statement

Our work will not cause ethical issues and the datasets used in this paper are publicly available.

References

Marco Ancona, Enea Ceolini, Cengiz Öztireli, and Markus Gross. 2018. Towards better understanding of gradient-based attribution methods for deep neural networks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30-May 3, 2018, Conference Track Proceedings*. OpenReview. net.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140.

David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. 2010. How to explain individual classification decisions. *The Journal of Machine Learning Research*, 11:1803–1831.

Jasmijn Bastings and Katja Filippova. 2020. The elephant in the interpretability room: Why use attention as explanation when we have saliency methods? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 149–155, Online. Association for Computational Linguistics.

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

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On identifiability in transformers. In *8th International Conference on Learning Representations (ICLR 2020)(virtual)*. International Conference on Learning Representations.

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

Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs. *URL https://www. kaggle. com/c/quora-question-pairs*.

Nicola De Cao, Michael Sejr Schlichtkrull, Wilker Aziz, and Ivan Titov. 2020. How do decisions emerge across layers in neural models? interpretation with differentiable masking. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3243–3255, Online. Association for Computational Linguistics.

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

Shuoyang Ding, Hainan Xu, and Philipp Koehn. 2019. Saliency-driven word alignment interpretation for neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 1–12, Florence, Italy. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Chaoyu Guan, Xiting Wang, Quanshi Zhang, Runjin Chen, Di He, and Xing Xie. 2019. Towards a deep and unified understanding of deep neural models in nlp. In *International Conference on Machine Learning*, pages 2454–2463. PMLR.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not Explanation. 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)*, pages 3543–3556, Minneapolis, Minnesota. Association for Computational Linguistics.

Joseph D Janizek, Pascal Sturmfels, and Su-In Lee. 2021. Explaining explanations: Axiomatic feature interactions for deep networks. *Journal of Machine Learning Research*, 22:1–54.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262, New Orleans, Louisiana. Association for Computational Linguistics.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 681–691, San Diego, California. Association for Computational Linguistics.

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2022. Less is more: Task-aware layer-wise distillation for language model compression. *arXiv preprint arXiv:2210.01351*.

Chang Liu, Chongyang Tao, Jiazhan Feng, and Dongyan Zhao. 2022. Multi-granularity structural knowledge distillation for language model compression. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1001–1011, Dublin, Ireland. Association for Computational Linguistics.

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

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 14037–14047.

Ali Modarressi, Hosein Mohebbi, and Mohammad Taher Pilehvar. 2022. AdapLeR: Speeding up inference by adaptive length reduction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–15, Dublin, Ireland. Association for Computational Linguistics.

Geondo Park, Gyeongman Kim, and Eunho Yang. 2021. Distilling linguistic context for language model compression. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 364–378, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Damian Pascual, Gino Brunner, and Roger Wattenhofer. 2021. Telling BERT's full story: from local attention to global aggregation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 105–124, Online. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

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

Karl Schulz, Leon Sixt, Federico Tombari, and Tim Landgraf. 2020. Restricting the flow: Information bottlenecks for attribution. *arXiv preprint arXiv:2001.00396*.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Sandipan Sikdar, Parantapa Bhattacharya, and Kieran Heese. 2021. Integrated directional gradients: Feature interaction attribution for neural NLP models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 865–878, Online. Association for Computational Linguistics.

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

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for BERT model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4323–4332, Hong Kong, China. Association for Computational Linguistics.

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

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

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International Conference on Machine Learning*, pages 3319–3328. PMLR.

Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: On the importance of pre-training compact models. *arXiv preprint arXiv:1908.08962*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: a stickier benchmark for general-purpose language understanding systems. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 3266–3280.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pre-trained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. 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)*, pages 11–20, Hong Kong, China. Association for Computational Linguistics.

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

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. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1355–1362, Online. Association for Computational Linguistics.

Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *13th European Conference on Computer Vision, ECCV 2014*, pages 818–833. Springer Verlag.

Quanshi Zhang, Xu Cheng, Yilan Chen, and Zhefan Rao. 2022. Quantifying the knowledge in a dnn to explain knowledge distillation for classification. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (01):1–17.

A Experimental Details

A.1 Details of Datasets

We evaluate AD-KD on eight tasks of GLUE benchmark (Wang et al., 2018). Specifically, there are two single-sentence tasks: CoLA (Warstadt et al., 2019) which aims to predict if the given sentence is grammatically correct, and SST-2 (Socher et al., 2013) which aims to predict the sentiment of the given sentence; two paraphrase tasks: MRPC (Dolan and Brockett, 2005) which aims to predict if two given sentences are semantically equivalent, and QQP (Chen et al., 2018) which is similar to MRPC; three inference tasks which aim to predict if the premise entails the hypothesis: MNLI (Williams et al., 2018), QNLI (Rajpurkar et al., 2016), and RTE (Bentivogli et al., 2009); and one similarity task: STS-B (Cer et al., 2017) which aims to predict a continual score measuring the semantic similarity between a pair of sentences. The statistics of these datasets are shown in Table 4.

Task	#Train	#Dev	#Test	#Label
Single-Sentence Classification				
CoLA	8.5k	1k	1k	2
SST-2	67k	872	1.8k	2
Pairwise Text Classification				
MNLI	393k	20k	20k	3
QNLI	108k	5.7k	5.7k	2
MRPC	3.7k	408	1.7k	2
QQP	364k	40k	391k	2
RTE	2.5k	276	3k	2
Text Similarity				
STS-B	7k	1.5k	1.4k	1

Table 4: Statistics of the GLUE datasets.

A.2 Hyperparameter Settings

We run all experiments on GeForce RTX 2080 Ti GPUs. Table 5 presents the hyperparameter settings and training costs of AD-KD on GLUE tasks. Generally, AD-KD runs 1.2 to 3 times slower compared to vanilla KD on different tasks, due to the extra back-propagation. However, all students obtained by different distillation methods have the same inference speed.

Hyperparameter	CoLA	MNLI	SST-2	QNLI	MRPC	QQP	RTE	STS-B
Learning Rate	4e-5	4e-5	5e-5	4e-5	3e-5	4e-5	2e-5	5e-5
Total Batch Size	32	64	32	32	16	32	16	16
Max Seq. Length	128	128	128	128	128	128	128	128
α	0.9	0.8	0.8	0.9	0.9	1.0	0.9	0.8
β	1	10	1	100	10	50	10	1
τ	1	3	2	3	4	4	2	3
K	768	768	768	768	768	734	700	640
m	1	1	1	1	1	1	1	1
# GPU	1	4	1	1	1	1	1	1
Training Time	30min	12hr	2.5hr	3hr	20min	16hr	12min	20min

Table 5: Hyperparameter settings and training cost.

B More Discussion

In this section, we discuss the difference between distilling the attribution maps and distilling the attention matrices. In a sense, attention matrices are similar to attribution maps since they both reflect the contribution that each input token makes on a model prediction to some extent (Bastings and Filippova, 2020; Xu et al., 2020). However, there are several drawbacks when it comes to distillation. On one hand, attention correlates well with attribution locally in specific layers and heads but not globally, indicating that attention maps are inadequate to draw conclusions that refer to the input of the model (Pascual et al., 2021). In other words, attention matrices are more like model-specific knowledge that are probably challenging for the student to learn due to the layer mapping issue, especially when the student has much fewer parameters than the teacher. On the other hand, some works point out that by adversarial training, alternative attention weights can be found whereas the prediction remains almost the same (Jain and Wallace, 2019; Wiegreffe and Pinter, 2019). Therefore, an optimal student unnecessarily shares similar attention matrices with its teacher. Our proposed AD-KD adopts a more reliable gradient-based method to obtain the attribution maps, which is shown better than attention matrices employed by baselines.

C More Experimental Results

C.1 Results on MultiRC

Considering that the text in GLUE is relatively short (with Max_Seq_Length set to 128), We conduct additional experiments on SuperGLUE (Wang et al., 2019) for more comprehensive evaluation. We select a challenging QA task, MultiRC (Khashabi et al., 2018), with much longer text (with Max_Seq_Length set to 512) which requires more attribution knowledge. As shown in Table 6, AD-KD improves 0.97% over vanilla KD and 0.38% over MGSKD. Moreover, the performance of AD-KD is on par with the teacher.

Model	#Params	Acc
BERT$_{base}$ (Teacher)	110M	68.53
Vanilla KD	66M	67.70
MGSKD	66M	68.29
AD-KD	66M	68.67

Table 6: Results on MultiRC development set.

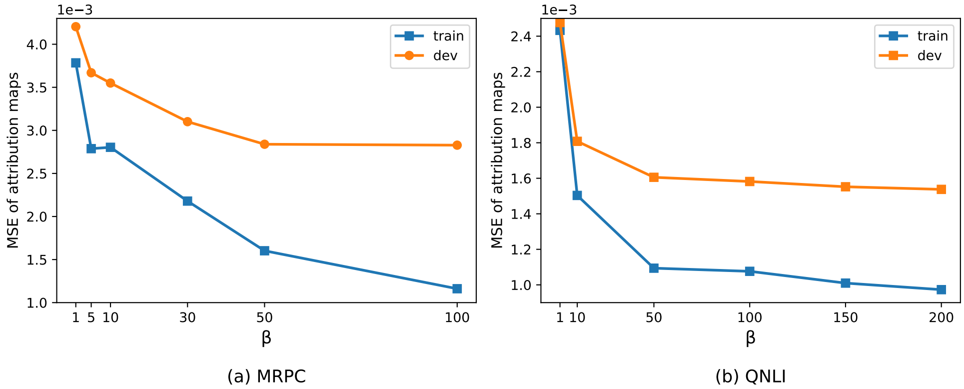

Figure 9: Comparison of the attribution gap between teacher and student on training set and development set.

C.2 Overfitting Study

In this section, we investigate whether the overfitting problem would happen in attribution distillation. Using Eq. (11), we calculate the attribution gap between the teacher and the student models on the training and development sets of MRPC and QNLI respectively, and show the results in Figure 9. By altering β, the tendency of attribution gap on development sets is consistent with the one on training sets, which indicates that the attribution knowledge learned from training data can be well generalized to unseen data. Therefore, overfitting tends not to happen in attribution distillation.

C.3 Case Study

In this section, we provide two examples to show how AD-KD facilitates the imitation of the teacher's reasoning and outperforms vanilla KD. As shown in Figure 10, vanilla KD makes mistakes by ignoring keyword *Louisiana* or emphasizing an irrelevant word *billion*. In contrast, the attribution maps of AD-KD are more consistent with the ones in the teacher. AD-KD learns what to and not to focus on and thus predicts the label correctly.

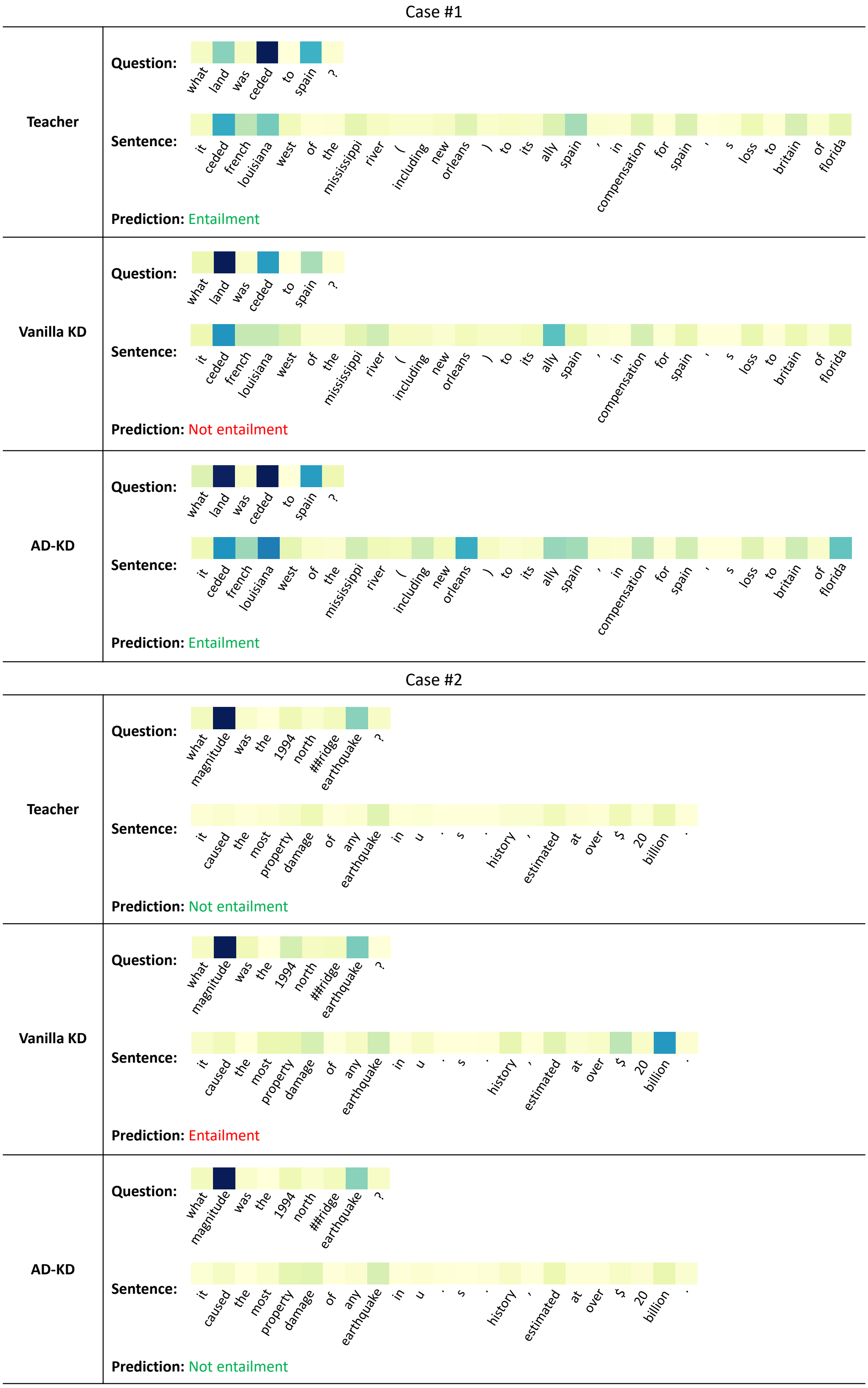

Figure 10: Two illustrative examples of attribution maps and predictions by teacher, vanilla KD and AD-KD from the QNLI development set, where darker colors mean larger attribution scores. In case #1, AD-KD learns which tokens to focus on (*Louisiana*), while in case #2, AD-KD learns which tokens not to focus on (*billion*).

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☑ A2. Did you discuss any potential risks of your work?
Section Ethics Statement

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section Abstract and Section 1

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

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 4.1 and Section 4.3

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
The datasets we use are consistent with those used in previous works.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 4.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix A.1

C ☑ Did you run computational experiments?

Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix A.2

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4.3 and Appendix A.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5.1

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4.3

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

Left blank.

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

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

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

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

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