



Explanation Guided Knowledge Distillation for Pre-trained Language Model Compression

ZHAO YANG and YUANZHE ZHANG, School of Artificial Intelligence, University of Chinese Academy of Sciences, China and The Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China

DIANBO SUI, School of Computer Science, Harbin Institute of Technology at Weihai, China

YIMING JU, JUN ZHAO, and KANG LIU, School of Artificial Intelligence, University of Chinese Academy of Sciences, China and The Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China

Knowledge distillation is widely used in pre-trained language model compression, which can transfer knowledge from a cumbersome model to a lightweight one. Though knowledge distillation based model compression has achieved promising performance, we observe that explanations between the teacher model and the student model are not consistent. We argue that the student model should study not only the predictions of the teacher model but also the internal reasoning process. To this end, we propose Explanation Guided Knowledge Distillation (EGKD) in this article, which utilizes explanations to represent the thinking process and improve knowledge distillation. To obtain explanations in our distillation framework, we select three typical explanation methods rooted in different mechanisms, namely *gradient-based*, *perturbation-based*, and *feature selection* methods. Then, to improve computational efficiency, we propose different optimization strategies to utilize the explanations obtained by these three different explanation methods, which could provide the student model with better learning guidance. Experimental results on GLUE demonstrate that leveraging explanations can improve the performance of the student model. Moreover, our EGKD could also be applied to model compression with different architectures.

CCS Concepts: • **Computing methodologies** → **Natural language processing**;

Additional Key Words and Phrases: Explanation, knowledge distillation, model compression

This work was supported by the National Key R&D Program of China (2022YFF0711900) and the National Natural Science Foundation of China (No. 61831022, No.62276264, and No. 62306087). This work is also supported by Yunnan Provincial Major Science and Technology Special Plan Projects (No.202202AD080004) and the Youth Innovation Promotion Association CAS. And this work is also supported by the Natural Science Foundation of Shandong Province (Grant No. ZR2023QF154). Authors' Addresses: Z. Yang, Y. Zhang (Corresponding author), Y. Ju, J. Zhao, and K. Liu (Corresponding author), School of Artificial Intelligence, University of Chinese Academy of Sciences, China and The Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, No. 95, Zhongguancun East Road, Beijing, China, 100190; e-mails: {zhao.yang, yuanzhe.zhang, yiming.ju, jzhao, kliu}@nlpr.ia.ac.cn; D. Sui, School of Computer Science, Harbin Institute of Technology at Weihai, No. 95, Zhongguancun East Road, Beijing, China, 100190; e-mail: suidianbo@hit.edu.cn.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://permissions.acm.org).

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM 2375-4699/2024/02-ART32

<https://doi.org/10.1145/3639364>

ACM Reference Format:

Zhao Yang, Yuanzhe Zhang, Dianbo Sui, Yiming Ju, Jun Zhao, and Kang Liu. 2024. Explanation Guided Knowledge Distillation for Pre-trained Language Model Compression. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 23, 2, Article 32 (February 2024), 19 pages. <https://doi.org/10.1145/3639364>

1 INTRODUCTION

Large pre-trained language models, like BERT [9] and GPT [24], have achieved cutting-edge results on various NLP tasks [6]. However, these models often involve billions or even trillions of parameters and thus have high latency, prohibitive memory footprint, and massive power consumption in application [13]. Therefore, lots of studies have explored compressing an original cumbersome model into a lightweight model without performance compromising [27, 33, 45].

One of the typical compression methods is knowledge distillation [14], which trains a lightweight model (Student) to emulate a cumbersome one (Teacher) by matching their predictions. However, solely matching the teacher's predictions cannot ensure the student model learns well from the teacher model [33]. To augment the vanilla knowledge distillation, recent studies [1, 16, 33] align not only the predictions in the output layer but also the internal representations.

Human teachers facilitate the development of the reasoning ability of students by requiring students to explain and show their thinking process [3, 40]. Inspired by this, we expect the lightweight student model in knowledge distillation should exhibit the same internal logic as the cumbersome teacher model. In this article, we leverage explanations¹ to reveal the internal logic of the model. Naturally, we need to investigate whether the current student models have a similar internal logic to the teacher model. Thus, we conduct experiments to analyze these student models. As shown in Figure 1, when we apply the widely used gradient explanation method [29] on both models in a sentiment analysis task, we could obtain explanation E_T and E_S of the teacher model and the student model, respectively. In specific, from E_T we could observe that the teacher model predicts this sentence as positive because of the presence of *exists* and *fine*. However, the student model relies on *and*, *its*, and *fine*, which is different from the teacher model. This difference suggests that the models may follow different reasoning processes. Furthermore, we perform a quantitative analysis with the same explanation method to obtain the explanations on the sentiment classification task (SST-2). From the results in Table 1, we observe that though knowledge distillation could improve performance, the explanations are quite different between the teacher model and the student model. Even armed with internal representations [33], their explanations still remain inconsistent. These results indicate that current compression methods can not exploit the full potential of knowledge distillation, and the knowledge maintained in the teacher model is not completely transferred into the student model. As a result, the student model could not perform well on both in-distribution and out-of-distribution tests.

To fully exploit the potential of knowledge distillation, we propose **Explanation Guided Knowledge Distillation (EGKD)**, which constrains the explanations of the student model to be consistent with the teacher model. Specifically, we utilize three kinds of well-explored explanation methods to obtain explanations, namely *gradient-based*, *perturbation-based*, and *feature selection* explanation methods. According to the different characteristics of these explanation methods, we design different ways to integrate them into a unified knowledge distillation framework, with the goal of effectively transferring the knowledge from the teacher model to the student model. Besides, compared with the gradient-based explanation methods, perturbation-based and feature

¹In this article, explanations refer to attribution scores contributing to the prediction for tokens in the input.

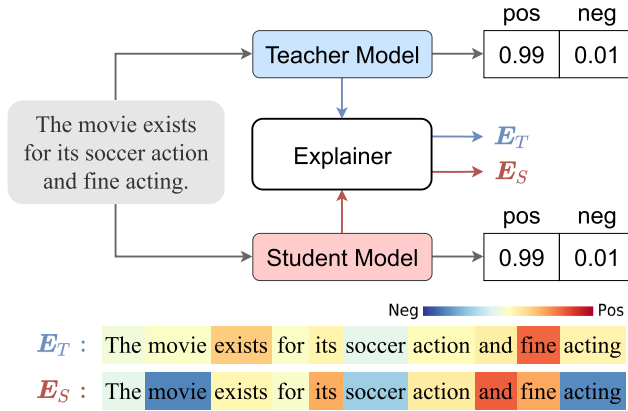


Fig. 1. The teacher and the student model predict the same because knowledge distillation constrains similar logits. However, when we use the gradient explanation method [29] to explore the attribution of each token contributes to the prediction, we can find that the obtained explanations E_T , E_S are obviously different.

Table 1. BERT₆-FT, BERT₆-KD Refer to the Naive Fine-tuning and the Vanilla Knowledge Distillation of 6-layer BERT, Respectively

Model	Spear \uparrow	Jac@50% \uparrow	Accuracy
BERT-base (Teacher)	1.00	1.00	93.8
BERT ₆ -FT	0.17	0.19	91.8
BERT ₆ -KD [14]	0.22	0.21	92.9
BERT ₆ -PKD [33]	0.22	0.22	93.1

BERT₆-PKD aligned hidden states as a more strict constraint. **Accuracy** refers to their performance on the test dataset. The other two metrics both evaluate the similarity of explanations between the teacher model and the student model. **Spear** shows the Spearman Correlation between the full attribution scores. **Jac50%** shows the Jaccard similarity of tokens that have top 50% attributions.

selection explanation methods both have high computation complexity and make knowledge transfer very time-consuming [30]. In specific, perturbation-based explanation methods need to sample many perturbed examples from the original input [19, 26], and feature selection methods require reparameterization and extra training [4, 15]. To accelerate the knowledge transfer process, we further introduce our novel optimization strategy for different explanation methods (Section 3) to improve efficiency. Experimental results conducted on GLUE [37] indicate that EGKD can achieve better performance and our efficiency optimization is effective.

Beyond obtaining good results on GLUE, we also claim that EGKD provides a more general approach to knowledge transfer than previous studies, which are featured by utilizing internal representations [16, 33]. These studies are all limited by the homogeneous assumption that the student model shares the same architecture as the teacher model. However, EGKD can fully relax this assumption since the explanation is independent of the model architecture and is only related to the input. Therefore, EGKD can be applied to heterogeneous model compression, like distilling a Transformer-based model into a BiLSTM model, which is easier deployed on resource-constrained mobile devices [10, 13].

The contributions can be summarized as follows:

- This article proposes explanation guided knowledge distillation (EGKD), which is the first work to introduce the explanation constraints into knowledge distillation based model compression.
- To improve knowledge transfer efficiency, this article proposes different optimization strategies to utilize explanations according to the different mechanisms of typical explanation methods.
- Various experimental results show EGKD performs well in both in-distribution and out-of-distribution conditions. Furthermore, EGKD is decoupled from the model architecture and can be applied to heterogeneous model compression.

The remainder of this article is structured as follows: Section 2 briefly reviews knowledge distillation and its recent application in model compression formally. Section 3 presents the details of EGKD. Section 4 describes the experimental settings and corresponding results. Additionally, we provide more detailed discussions of the proposed methods in Section 5 and present relevant background information in Section 6. Finally, Section 7 concludes the article.

2 BACKGROUND

In this section, we introduce the basic information of knowledge distillation in Section 2.1 and recent studies that utilize internal representations to augment knowledge distillation in Section 2.2.

2.1 Vanilla Knowledge Distillation

Knowledge distillation [14] is widely used in model compression, which encourages the student model f_S to mimic the teacher model f_T via matching their logits. Formally, for a K -classes classification task and the input x_i , the loss of matching their logits can be computed as:

$$L_{logit} = \sum_i \sum_{k \in K} [\text{softmax}(f_T(x_i)/T) \cdot \log(\text{softmax}(f_S(x_i)/T))] \quad (1)$$

where softmax refers to the softmax operation, T is the temperature of knowledge distillation, which adjusts the scale of the logit. And the common cross entropy loss can be computed as:

$$L_{CE} = \sum_i \sum_{k \in K} [\mathbb{I}[y_i = k] \cdot \log P(y_i = k|x_i)] \quad (2)$$

where \mathbb{I} is an indicator function and y_i is the label of x_i . Besides, $P(y|x)$ is the equivalent form of f .

In summary, for vanilla knowledge distillation, the corresponding total loss function is:

$$L_{KD} = \alpha L_{CE} + (1 - \alpha) L_{logit} \quad (3)$$

where $\alpha \in [0, 1]$ is the loss weight.

2.2 Knowledge Distillation with Internal Representations

We simply introduce recent studies [33] utilize the internal representations of the teacher model to enhance knowledge distillation. Suppose that the teacher and student models are both Transformer [36] with $M, N (M > N)$ layers, respectively. $N - 1$ layers should be selected from the teacher to match the first $N - 1$ layer of the student. The loss of matching hidden states is defined as:

$$L_{hidden} = \sum_{m=1}^{N-1} \text{MSE} \left(h_m^S, h_{I_{select}(m)}^T \right) \quad (4)$$

where h_m^S is the hidden state of the m -th layer of the student model, and $h_{I_{select}(m)}^T$ is the hidden state of the layer in the teacher model that matches the m -th layer of the student model. Specifically,


Explanation Method	Explanation	Neg  Pos
Gradient-based	the most purely enjoyable and satisfying evenings	
Perturbation-based	the most purely enjoyable and satisfying evenings	
Feature Selection	the most purely enjoyable and satisfying evenings	

Fig. 2. Explanations which are obtained by different explanation methods for a sentence in the SST-2 dataset.

the hidden state is the representation of the [CLS]. If we only match hidden states [1, 33], the total loss function is:

$$L_{KD} = \alpha L_{CE} + (1 - \alpha)L_{logit} + \beta L_{hidden} \quad (5)$$

where α , β is the loss weight.

Similarly, the loss of attention matrices is:

$$L_{att} = \sum_{m=1}^{N-1} \frac{1}{d} \sum_n \text{MSE} \left(A_{m,n}^S, A_{I_{select}(m,n)}^T \right) \quad (6)$$

where d is the number of attention heads; and $A_{m,n}^S, A_{m,n}^T$ refer to the attention matrix of the n -th head of the m -th layer of the student model and the teacher model, respectively. If we further match hidden states and attention matrices [16, 18] at the same time, the corresponding total loss function is defined as:

$$L_{KD} = \alpha L_{CE} + (1 - \alpha)L_{logit} + \beta(L_{hidden} + L_{att}) \quad (7)$$

where α , β is the loss weight.

3 EXPLANATION GUIDED KNOWLEDGE DISTILLATION

3.1 Overall Description of EGKD

For a given model f and a given sentence $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$, we can obtain the attribution value vector $\mathbf{a}_i = (a_{i,1}, a_{i,2}, \dots, a_{i,n})$ through an explanation method E . $a_{i,j}$ ² is the attribution value of token $x_{i,j}$, and refers to the contribution of $x_{i,j}$ to the prediction. This process can be denoted as:

$$\mathbf{a}_i = E(x_i, f(x_i), f) \quad (8)$$

where $f(x_i)$ denotes the prediction of x_i .

According to different mechanisms, current explanation methods can be classified as *gradient based*, *perturbation based*, and *feature selection* methods [15]. An example of explanations obtained by different explanation methods for a sentence in SST-2 is shown in Figure 2.

Let denote the explanations of the teacher and the student model as \mathbf{a}_i^T and \mathbf{a}_i^S , respectively. In the proposed EGKD, we require the student model to study not only the predictions of the teacher model but also explanations. That constraint on explanations could be uniformly modeled as:

$$L_{exp} = \sum_i \text{MSE} \left(\mathbf{a}_i^T, \mathbf{a}_i^S \right) \quad (9)$$

Following Equation (7), the corresponding total loss function is defined as:

$$L_{explanation} = \alpha L_{CE} + (1 - \alpha)L_{logit} + \beta L_{exp} \quad (10)$$

where α , β is the loss weight.

²For gradient-based and perturbation-based methods, $a_{i,j}$ is continuous value. For feature selection methods, $a_{i,j}$ is 0 or 1.

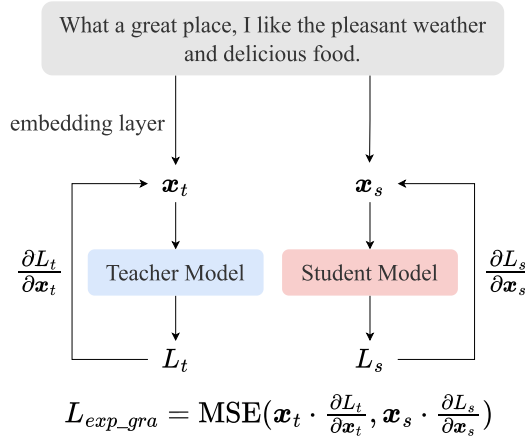


Fig. 3. Gradient-based explanation guided knowledge distillation. L^T, L^S refers to the logits of the teacher model and student model.

3.2 EGKD with Different Explanation Methods

However, different from the internal representations, it is time-consuming to obtain the explanations by existing explanation methods [30], especially for perturbation-based explanation methods and feature selection explanation methods. Therefore, if we generate explanations while training the model, the training time of EGKD would be greatly increased. In the following subsections, we will detail how to efficiently compute and incorporate the aforementioned explanation matching loss for different explanation methods.

3.3 Gradient-based Explanation Guided Knowledge Distillation

Gradient-based explanation methods compute $a_{i,j}$ via the gradient of the model [29]:

$$a_{i,j} = \frac{\partial L}{\partial \mathbf{x}_{i,j}} \cdot \mathbf{x}_{i,j} \quad (11)$$

where L is the loss of the model prediction. Some variations like Smooth Gradient [31] and Integrated Gradient [34] also follow this formula.

As shown in Figure 3, when we want to match the gradient-based explanations between the teacher and the student model, we can match the attribution scores in Equation (11). Both the gradient value $\frac{\partial L}{\partial \mathbf{x}_i}$ and the embedding \mathbf{x}_i are internal representations. Therefore, the loss function to match the gradient explanations between the teacher model and the student model can be formulated as follows:

$$L_{exp_gra} = \sum_i \sum_j \text{MSE} \left(\frac{\partial L^T}{\partial \mathbf{x}_{i,j}^T} \cdot \mathbf{x}_{i,j}^T, \frac{\partial L^S}{\partial \mathbf{x}_{i,j}^S} \cdot \mathbf{x}_{i,j}^S \right) \quad (12)$$

where L^T, L^S refers to the loss of the teacher model and the student model, respectively. And $\mathbf{x}_{i,j}^T, \mathbf{x}_{i,j}^S$ stand for the textual representation of $x_{i,j}$ for teacher model and student model, respectively.

3.4 Perturbation-based Explanation Guided Knowledge Distillation

Perturbation-based explanation methods first sample a binary mask vector $\mathbf{z}_i = (z_{i,1}, z_{i,2}, \dots, z_{i,n})$, where $z_{i,j}$ indicates whether $x_{i,j}$ is present ($z_{i,j} = 1$) or absent ($z_{i,j} = 0$). And $M_x(\mathbf{z})$ can map the

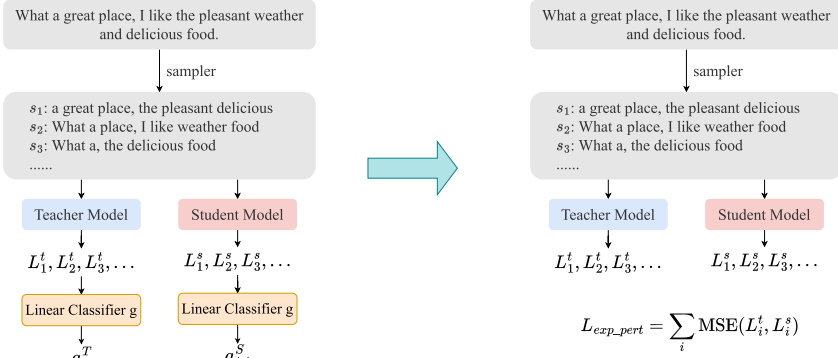


Fig. 4. Optimization of perturbation-based explanation guided knowledge distillation. On the left, $a_{i,j}^T, a_{i,j}^S$ stands for the explanation of the teacher model and the student model, respectively. On the right, L_i^T, L_i^S refers to the logits of the teacher model and student model for sentence x_i . Originally, we should compute losses with $a_{i,j}^T$ and $a_{i,j}^S$, to improve efficiency, we turn to utilize Equation (14).

mask z to the perturbed input x' . In summary, these methods seek to learn a local linear classifier g on z to align the prediction of model f [44]:

$$g(z) = c + \sum_{j=1}^n a_{i,j} z_{i,j} \quad (13)$$

$$a_{i,j} = \arg \min_g \sum_{z \in Z} \pi_x(z) [f(M_x(z)) - g(z)]^2$$

where $\pi_x(z)$ is a local kernel to assign weight to each perturbation z and Z is the set of perturbations. Specifically, LIME [26] sets $\pi_x(z)$ as an exponential kernel and Leave-One-Out [19] is a special case of LIME. SHAP [20] designs $\pi_x(z)$ so that the attribution can be seen as Shapley Values.

As shown in Figure 4, when we want to match the perturbation-based explanations between the teacher and the student model, we do not have to compute the attribution via Equation (13). The only difference between the teacher model and the student model is the f in Equation (13). Therefore, we can simplify the loss function as follows:

$$L_{exp_pert} = \sum_i \sum_{z \in Z} \text{MSE}(f_T(M_x(z)), f_S(M_x(z))) \quad (14)$$

3.5 Feature Selection Explanation Guided Knowledge Distillation

Feature selection explanation methods aim to find a minimal sufficient subset of the original inputs, which ensures these features alone suffice for the same prediction as the originals. To find a subset for $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,n})$, we always train a binary mask vector $z_i = (z_{i,1}, z_{i,2}, \dots, z_{i,j})$, where $z_{i,j}$ refers to whether $x_{i,j}$ should be reserved. And we also define $M_x(z_i)$ to map z_i to the masked input x'_i . To satisfy sufficiency, we should ensure the prediction difference is small enough. To satisfy minimal, we should try to make the size of the subset small enough. Therefore, we can get the explanation e_i for x_i as follows [4, 15, 17]:

$$e_i = \arg \min_{z_i \in Z} \lambda_1 L(f(x_i), f(M_x(z_i))) + \lambda_2 \sum_{j=1}^n z_{i,j} \quad (15)$$

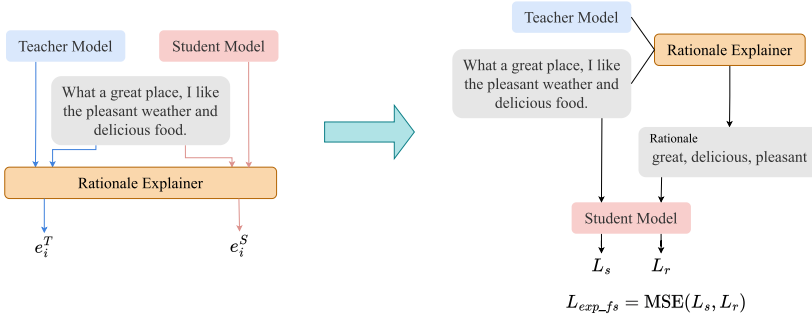


Fig. 5. Optimization of feature selection explanation guided knowledge distillation. On the left, e_i^T, e_i^S stands for the explanation of the teacher model and the student model, respectively. On the right, L_s, L_r means the logits of the student model when feeding the original sentence and the rationale of the teacher model, respectively. Originally, we should compute losses with e_i^T and e_i^S . To improve efficiency, we approximate the original loss function as Equation (18).

where Z denotes the set for all of possible binary mask vector z_i , and λ_1, λ_2 are the hyper-parameters for loss weights. The first term ensures sufficiency, and the second term computes the number of 1 in z_i , which can guarantee the size of the subset is small enough.

If we want to align the feature selection explanations between the teacher model and the student model, we just need to compute the following loss:

$$L_{x_i} = \text{Loss} \left(e_i^T, e_i^S \right) \quad (16)$$

Since e_i^T is fixed for a trained teacher model, if we want the student model to imitate the explanation of the teacher model, that means that we need to force $e_i^S = e_i^T$. Then we consider how we can make the explanation of the student model e_i^S approach e_i^T . Therefore, we approximate the above problem as follows:

$$L_{x_i} = \text{MSE} \left(f_S \left(M_x \left(e_i^T \right) \right), f_S(x_i) \right) \quad (17)$$

Optimizing Equation (17) is actually optimizing the first term in Equation (15). Now e_i^T satisfy sufficiency for the student model f_S . Therefore, we believe Equation (17) is an approximation of Equation (16). And we utilize this approximation in the following experiment. In summary, as shown in Figure 5, we can formulate the loss function to match the feature selection explanations between the teacher model and the student model as follows:

$$L_{exp_fs} = \sum_i \text{MSE} \left(f_S \left(M_x \left(e_i^T \right) \right), f_S(x_i) \right) \quad (18)$$

In summary, for the three typical explanation methods, we show the detail of how to compute the explanation matching loss in Equation (9). In our following experiments, we could replace L_{exp} with the specific explanation matching loss in Equation (10) to construct the whole loss function for these different variants of EGKD.

4 EXPERIMENTS

4.1 Experimental Data and Evaluation Metrics

We evaluate all of the models on the classification³ tasks of **General Language Understanding Evaluation (GLUE)** [37]. We select one text classification task: SST-2 [32], two sentence similarity

³We do not select the regression task STS-B because current explanation methods focus on the classification task.

Table 2. Statistics of the Datasets in GLUE

Dataset	Task Type	Train / Valid / Test	Metric
MNLI-m	Natural Language Inference	393k / 9.8k / 20k	Acc
MNLI-mm	Natural Language Inference	393k / 9.8k / 20k	Acc
MRPC	Sentence Similarity	3.7k / 0.4k / 1.7k	F1
QNLI	Natural Language Inference	105k / 5.4k / 5.4k	Acc
QQP	Sentence Similarity	364k / 40k / 391k	F1
SST-2	Single-sentence Classification	67k / 0.8k / 1.8k	Acc
RTE	Natural Language Inference	2.5k / 0.2k / 3k	Acc

tasks: MRPC [11], QQP [8], and three natural language inference tasks: MNLI [41], QNLI [25], RTE [5].⁴

Following previous works [16, 18], we use classification accuracy as the evaluation metric for MNLI-m, MNLI-mm, QNLI, RTE, and SST-2. And we use F1 metric for MRPC and QQP for fair comparisons. All of the results are reported on the test set of the GLUE. Table 2 shows the details of the evaluation datasets.

4.2 Implementation Details

Following the previous work [16], we use BERT-BASE [9] as the teacher model and select 6-layer and 4-layer BERT as the student models. The batch size is set to 16, the learning rate is set to $1e-5$ and the number of training epochs is set to 10. To determine the other hyperparameters, we employ a grid search algorithm on the validation set. In detail, we first tune the loss function weight α in $\{0.2, 0.5, 0.7\}$ and the temperature T in $\{1, 5, 10\}$, and we fix α and T to the values with best performance from vanilla KD experiments. Then we only search the loss function weight β in $\{0.01, 0.005, 0.001\}$. All experiments are conducted with an NVIDIA GeForce RTX 3090 Ti.

4.3 Baselines

We compare the following **state-of-the-art (SoTA)** methods in the following experiments: (1) naive fine-tune, which refers to only fine-tuning the student model on the dataset, namely BERT₆/BERT₄-FT. (2) vanilla knowledge distillation, which is called BERT₆/BERT₄-KD (3) PKD [33] and the concurrent work [1] further leverage the internal representation matching based on the vanilla knowledge distillation. These baselines are called BERT₆/BERT₄-PKD. (4) TinyBERT [16] and BERT-EMD [18] introduce the hidden states matching and attention matrix matching in knowledge distillation.⁵ We name these baselines as BERT₆/BERT₄-PKD + attention.

4.4 Experimental Results on GLUE

We submitted the model predictions to the official GLUE evaluation server to obtain the results on the test set and Table 3 shows the detailed results. Overall, the experiment results from the 4-layer or the 6-layer student models consistently demonstrate that EGKD can achieve better performance than the baseline methods.

In detail, we find that: (1) For the 6-layer student model, compared to the best baseline, the best variant of EGKD improves 0.29% average scores on GLUE. Especially on the RTE dataset, our proposed method obtains a 1.3% improvement over the best baseline. For the 4-layer student

⁴We do not select CoLA which tests whether a sentence is grammatical. Many errors are due to the lack of components and explanation methods explore what parts of the input lead to the prediction, therefore they are not suitable for CoLA.

⁵To conduct a fair comparison, we just utilize the task distillation in TinyBERT and do not apply the general distillation.

Table 3. Results on GLUE

Model	Params Num	MNLI-m	MNLI-mm	MRPC	QNLI	QQP	SST-2	RTE	AVG
BERT-BASE(Teacher)	110M	84.6	83.5	86.4	90.6	71.0	93.8	67.4	82.47
BERT-BASE(Teacher)*	110M	83.9	83.4	87.5	90.9	71.1	93.4	67.0	82.46
BERT ₆ -FT	66M	82.2	81.1	82.8	89.0	70.0	92.6	59.1	79.54
BERT ₆ -KD*	66M	80.2	79.8	86.2	88.3	70.1	91.5	64.7	80.11
BERT ₆ -KD	66M	82.7	81.7	85.7	89.2	70.3	92.9	62.8	80.76
<i>Internal Representation</i>									
BERT ₆ -PKD*	66M	81.5	81.0	85.0	89.0	70.7	92.0	65.5	80.67
BERT ₆ -PKD	66M	83.0	81.8	86.0	89.0	70.4	93.1	63.0	80.90
BERT ₆ -PKD + attention	66M	83.0	82.3	86.1	89.5	70.4	93.1	63.1	81.07
<i>Explanation</i>									
BERT ₆ -EGKD _{gra} (ours)	66M	83.1	82.0	86.1	89.5	71.0	93.1	63.1	81.13
BERT ₆ -EGKD _{pert} (ours)	66M	83.5	82.3	86.3	89.7	70.3	93.1	64.3	81.36
BERT ₆ -EGKD _{fs} (ours)	66M	83.2	82.1	86.0	89.5	70.3	93.1	63.8	81.14
BERT ₄ -FT	52.2M	79.8	79.5	83.5	86.9	69.4	90.6	62.8	78.93
BERT ₄ -KD	52.2M	81.5	79.6	85.2	87.8	69.8	91.4	62.3	79.66
<i>Internal Representation</i>									
BERT ₄ -PKD*	52.2M	79.9	79.3	82.6	85.1	70.2	89.4	62.3	78.40
BERT ₄ -PKD	52.2M	81.3	80.0	84.3	87.8	69.7	90.8	64.4	79.75
BERT ₄ -PKD + attention	52.2M	81.3	80.0	84.2	87.8	69.9	91.0	64.4	79.80
<i>Explanation</i>									
BERT ₄ -EGKD _{gra} (ours)	52.2M	81.5	80.1	85.5	88.2	69.2	91.0	63.8	79.90
BERT ₄ -EGKD _{pert} (ours)	52.2M	81.6	80.1	85.8	88.0	70.0	91.5	63.8	80.11
BERT ₄ -EGKD _{fs} (ours)	52.2M	81.3	80.1	85.4	87.9	69.7	91.3	63.8	79.93

All results are reported from the test set of GLUE benchmark. We split the results for 6-layer BERT and 4-layer BERT in the table. Results with * refer to the results in the original article. For fair comparisons, we reproduce all of the baselines and show the results in the table. Actually, compared to the original article, all of our reproduced baselines get better performance on GLUE.

model, the best variant of EGKD gets 0.31 improvement on average scores of GLUE. On MRPC, it increases 1.5% in F1 scores. (2) Among the three variants, EGKD_{pert} gets the best performance. This variant outperforms the other variants 0.22% and 0.18% average scores on GLUE for 6-layer and 4-layer students, respectively.

Besides achieving the best performance on GLUE, EGKD_{pert} could also be applied in more scenarios. In detail, in the black-box scenarios, EGKD_{gra} is not applicable because we cannot compute the gradient. If we cannot compute explanations in advance, EGKD_{fs} also does not work. Therefore, among the proposed variants, EGKD_{pert} is the better choice in practice.

4.5 Beyond Preserve Accuracy— OOD Test and Loyalty Test

Current evaluation for model compression always tests the compressed model on the same test dataset. Actually, the ultimate goal of model compression is not just to perform well under the same test set [43]. Inspired by [43], we further perform **OOD (out-of-distribution)** test and loyalty test. OOD test could check the generalization of the model, and loyalty test could check the similarity of the outputs between the teacher model and the student model.

We select the 6-layer BERT model as the student model and utilize the well-trained model to perform these two tests. Specifically, we conduct experiments on MNLI and we choose the test dataset of HANS as the corresponding OOD test dataset. For the loyalty test, we measure the similarity of the output labels and the output probabilities.

Table 4. Results of OOD Test and Loyalty Test

Model	OOD Test		Loyalty Test	
	Ori	OOD	Label	Probability
BERT-BASE	84.5	59.8	100.0	100.0
BERT ₆ -KD	82.7	58.3	88.2	93.2
BERT ₆ -PKD	83.0	54.6	88.2	93.3
BERT ₆ -PKD + attention	83.0	47.7	88.9	93.5
BERT ₆ -EGKD _{gra}	83.1	52.3	89.4	93.8
BERT ₆ -EGKD _{pert}	83.5	58.7	89.6	94.0
BERT ₆ -EGKD _{fs}	83.2	55.4	89.4	93.8

We select 6-layer BERT model as the student model. And we just utilize the well-trained model on MNLi-m to perform these two tests.

Table 4 presents the corresponding results. In the OOD test, we find existing methods which utilize internal representations even perform worse on the OOD dataset while getting better performance on the in-distribution dataset. We conjecture the reason behind that is the more constraints may lead to the student model overfit the original datasets, thus make these student model perform not well on the OOD dataset. In contrast, all EGKD variants get better performance on OOD dataset than existing methods. In the loyalty test, we find the similarity increases with more constraints in existing methods. And EGKD also gets better output similarity compared to existing methods. Besides, in both tests, EGKD_{pert} gets the best performance among the three EGKD variants.

In summary, under more evaluation criteria [43] for model compression, our proposed EGKD also gets better performance compared to existing methods. And EGKD_{pert} also achieves the best performance in these evaluations.

4.6 Verification of Efficiency Optimization

To illustrate our optimization in the Section 3 could save time, we take Equation (9) as the baselines, which generates explanations for the student model when training. Figure 6 shows the corresponding cost time for these three variants, in which we show the training time of each epoch and the time of generating explanations for the teacher model. According to the figure, we can observe that our optimization could save much time, which verifies our optimization is useful.

In specific, for EGKD_{gra}, since the step of computing gradient is hard to save time, we do not make extra optimization. Thus, the training time of our method is almost equal to the training time before optimization. However, the time of generating explanations could be saved. For EGKD_{pert}, the training time is limited by the GPU memory size. We show the training time when setting the batch size 1 and 2. The training time would be further decreased with a bigger GPU memory size. For EGKD_{gra} and EGKD_{pert}, we also do not need extra time to generate explanations for the teacher in advance, which makes these two variants could be applied to more scenarios compared to no optimization version. As for EGKD_{fs}, the time of generating explanations could not be saved. However, the training time is reduced 1,000 times compared to its no-optimization version.

To illustrate the effectiveness of our optimization, we also test the performance before and after optimization and Table 5 presents the results. For EGKD_{gra} and EGKD_{pert}, the optimization does not sacrifice performance. For EGKD_{fs}, because of the approximation in Equation (17), the performance decreases only 0.2%. We consider this small performance sacrifice to be acceptable compared to the more than 1,000-times efficiency gain.

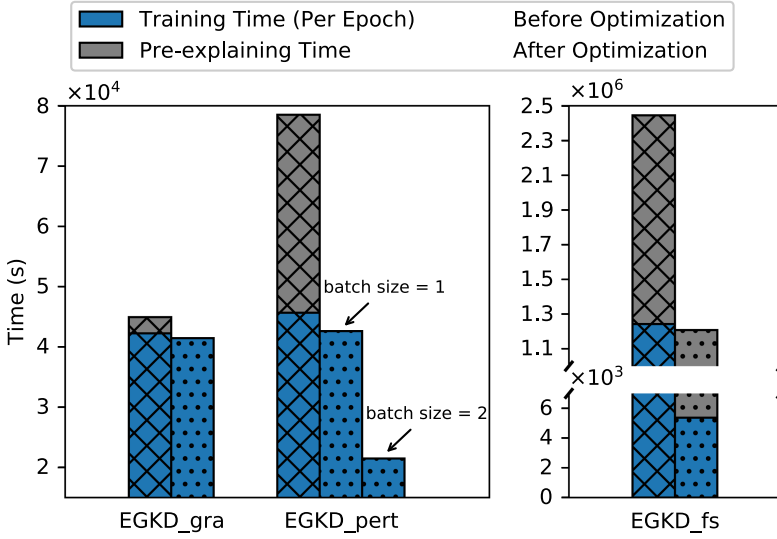


Fig. 6. Time analysis of the three variants on SST-2 when distilling into 4-layer BERT. After our optimization, the training time has decreased, especially on perturbation-based explanation methods and feature selection explanation methods.

Table 5. Performance Comparisons on SST-2 before and after Efficiency Optimization when Distilling into 4-layer BERT

EGKD _{gra} (w/o optimization)	EGKD _{gra}	EGKD _{pert} (w/o optimization)	EGKD _{pert} (batch size = 1)	EGKD _{pert} (batch size = 2)	EGKD _{fs} (w/o optimization)	EGKD _{pert}
91.0	91.0	91.5	91.5	91.5	91.3	91.1

4.7 Knowledge Distillation for Different Model Architectures

Existing methods mainly utilize the internal representation of the teacher model, which requires the student model shares a similar architecture with the teacher model. EGKD just utilizes the explanations, which are only related to the input. Therefore, different from the mainstream methods, EGKD can also be applied to knowledge distillation with different model architectures.

Following previous work [35], we select BiLSTM as the student model and BERT-base as the teacher model. We choose the same settings of BiLSTM with [35] and use 300-dimensional pre-trained GloVe word embeddings [22]. We conduct experiments on SST-2, QQP, and MNLI.

Table 6 presents the corresponding results. When compressing BERT-base into BiLSTM, the model size could reduce nearly 20 times. Besides, the inference time also decreases a lot. Specifically, compared to compressing BERT into a 6-layer or 4-layer, which only reduces the inference time 2× and 3×, compressing it into BiLSTM could reduce the inference time nearly 20×. Compared to vanilla knowledge distillation, the three variants of EGKD can further improve the performance on the whole datasets. Especially for EGKD_{pert}, this variant gets the best performance, which achieves improvements from 0.6% to 1.0% among these four datasets compared to the vanilla knowledge distillation.

5 DISCUSSION

5.1 Relation between Internal Representations and Explanations

In previous experiments, the internal representations and explanations are used in knowledge distillation alone. Thus, we naturally want to know whether the internal representations could work

Table 6. Results of Compressing BERT-base into Simple Neural Network BiLSTM

Model	Params	SST-2	QQP	MNLI-m/mm
BERT-BASE	110M	93.8	71.0	84.6/83.5
BiLSTM	5.3M	86.0	62.3	67.9/67.4
BiLSTM-KD	5.3M	87.1	63.6	69.4/68.3
BiLSTM-EGKD _{gra}	5.3M	87.5	64.1	69.7/68.8
BiLSTM-EGKD _{pert}	5.3M	87.8	64.2	70.1/69.3
BiLSTM-EGKD _{fs}	5.3M	87.4	63.9	69.9/69.0

We test on SST-2, QQP, and MNLI and we show the results in this table.

Table 7. Results of GLUE for 6-layer BERT which Combining Explanations with Internal Representations

Model	MNLI-m/mm	MRPC	QNLI	QQP	SST-2	RTE	AVG
EGKD _{pert}	83.5/82.3	86.3	89.7	70.3	93.1	64.3	81.36
+hidden	83.6/82.3	86.5	89.7	70.7	93.1	64.5	81.48
+hidden+att	83.6/82.4	86.5	90.1	70.6	93.1	64.6	81.56

together with the explanations in knowledge distillation. Then we explore whether the performance would be further improved when combining explanations with internal representations. Specifically, we add the constraints of internal representations on EGKD.

Table 7 presents that the performances of EGKD can be further improved. From these results, we could observe that EGKD_{pert}, the variant which gets the best performance, can be improved when aligning the extra hidden states. And the performance also gets further improved when adding the additional constraint of aligning both hidden states and attention matrices. These results indicate that matching explanations and internal representations could work together to improve the student, which means that explanations and internal representations are complementary to some extent. Therefore, when the student model has a similar architecture to the teacher model, we can add the constraints of internal representations on EGKD, which can further improve the performance of student model.

5.2 Exploration of Different Variants of EGKD

We propose three variants of EGKD in the previous sections and only list the performance of each of the three. In this subsection, we further analyze the proposed three variants of EGKD. Specifically, in our previous experiments, we leverage the formulated explanation matching losses of the different explanation methods to replace L_{exp} in Equation (10). We naturally want to know whether these different explanation matching losses can be combined and how the corresponding performance would change. Thus, we combine these losses to replace L_{exp} in Equation (10) and conduct experiments on GLUE for the 6-layer student model.

Figure 7 shows the average performance on GLUE for these different variants. We can find the performance would decrease when we combine the loss of the gradient-based and the perturbation-based methods. The performance would be further improved when combining the loss of the feature selection methods with the loss of the gradient-based or the perturbation-based methods. Thus, when we combine the loss of the perturbation-based method and the feature selection method, the corresponding result gets the best performance but the performance would drop when combining the whole three losses. To illustrate these results, we go back to the characteristics of these three explanation methods. Gradient-based and perturbation-based methods both show the

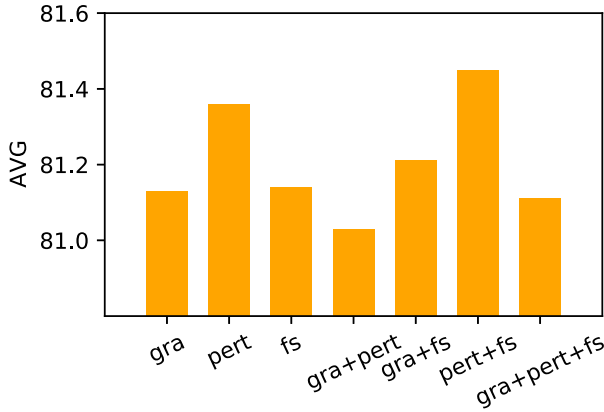


Fig. 7. Results of average performance on GLUE. We combine different losses of the three variants as the regularization term and show the corresponding results.

attribution scores for each token, and these methods have shown that they would give different scores for the same instance [2]. Therefore, when we combine these two losses, the student model would be confused to get close to which score. However, feature selection methods do not care about the specific score and only focus on the important tokens. Therefore, when we combine it with the loss of gradient-based or perturbation-based method, the student model can not only know what the important part is but also learn the specific attribution score for each token, which is consistent with [7]. Besides, perturbation-based methods have been shown that could get more faithful explanations compared to gradient-based methods. Thus, combining the loss of feature selection method with the loss of perturbation-based method could get better performance.

5.3 Effect of Sample Size for $EGKD_{\text{pert}}$

Among the proposed three variants, $EGKD_{\text{pert}}$ gets the best performance. In this variant, the sample size, which refers to $|Z|$ in Equation (14), is the most important parameter. Thus, we explore the effect of the sample size for $EGKD_{\text{pert}}$.

Specifically, we select 6-layer BERT as the student model and choose SST-2 as the target dataset. We show the performance of the valid dataset and the explanation similarity. And we also test the model performance on the test dataset of IMDB to explore the generalization. Figure 8 shows the corresponding results. We can observe that both the in-distribution and out-of-distribution performance are correlated with explanation similarity because of their same change trend. These results reveal the close relationship between the explanation similarity and the performance of the student model. Besides, we guess the best sample size differs among different datasets, which leads to different change trends. Too big sample size usually could not achieve the best performance because of too much noise. And if the sample size is small, the contribution of the perturbation-based methods would not be computed accurately, which could not obtain the best performance, either.

Furthermore, Equation (14), the final form of $EGKD_{\text{pert}}$, is similar with data augmentation, especially for the word-deletion based data augmentation methods [39]. However, the main motivation of $EGKD_{\text{pert}}$ is not to leverage more augmented data to help knowledge distillation. Our method $EGKD$ aims to better transfer the knowledge from the teacher model to the student model by matching the explanations between the teacher model and the student model. In specific, we hope the student model could learn the better attention to each token, which is generated by the teacher model. Especially for $EGKD_{\text{pert}}$, the explanations are generated by sampling many after-deletion

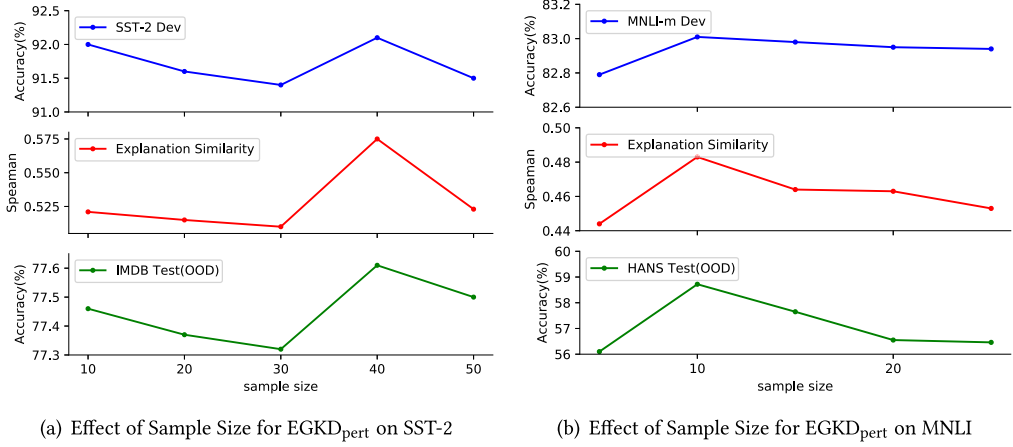


Fig. 8. Effect of sample size for BERT₆-EGKD_{pert} on SST-2 and MNLI. We show the performance on the dev set of the original dataset and the performance of corresponding OOD dataset. Besides, we also utilize Spearman Correlation Coefficient to evaluate the explanations similarity and list the similarity results.

sentences. Therefore, we think EGKD_{pert} can be seen as the theoretical support for the performance gain of word-deletion based data augmentation methods [39].

Moreover, we also do not think that the performance gain of EGKD_{pert} is totally from data augmentation. As most data augmentation methods show [12], the performance would improve with bigger augmented data size. However, as shown in Figure 8, the performance of EGKD_{pert} does not increase with bigger sample size. In our view, the explanation similarity is the more important reason for the performance of EGKD_{pert}.

5.4 Error Analysis

The teacher model usually could not get 100% accuracy among the test datasets. Therefore, its reasoning ways (explanation) would also lead to wrong answers. However, our explanation loss function encourages the student model to imitate the reasoning ways of the teacher model in these instances which the teacher model could not solve correctly. At the same time, the loss of ground truth encourages the model to predict the correct label in these instances, which is a contradiction. Therefore, the student model could not imitate all the reasoning ways of the teacher model, which leads to the gap between the teacher model and the student model. And we leave this problem as our future work to solve.

6 RELATED WORK

6.1 Pre-trained Language Model Compression

Pretrained language models have achieved promising performance on various NLP tasks, but they are trapped in application due to their high storage costs and massive power consumption. Current mainstream model compression techniques include weight quantization [28, 45], structure pruning [21, 38] and knowledge distillation [1, 16, 18, 33].

The weight quantization technique aims to reduce the number of bits needed to store weights. Most computer architectures use 32 bits to represent weights and existing quantization methods try to leverage fewer bits to store weights with less precision loss and performance sacrifice. Structure pruning based model compression technique aims to prune away structures like neurons,

attention heads, or layers. Existing methods try to find the most useless structures in the model and then prune them to reduce the model size. In this article, we just list these two mainstream techniques and only focus on knowledge distillation based model compression.

Knowledge distillation [14] is proposed to transfer the knowledge of a strong teacher model into the weak student model, which naturally fit the goal of model compression. In recent years, knowledge distillation based model compression methods focused on how to utilize the internal representation of the teacher model to help the student model learn better. [33] first proposed to utilize the hidden states ([CLS] representations) to help the student model learn more from the teacher model. They mapped some layers of the teacher model into each layer of the student model and the aligning the hidden states of these layers. The concurrent study [1] also proposed a similar method which leverages hidden states to enhance knowledge distillation. TinyBERT [16] performed both pre-training distillation and task distillation. In task distillation, they utilize both the [CLS] representations and the attention matrices. BERT-EMD [18] follow the fine-tuning distillation of TinyBERT, but they do not specify the correspondence between layers. They proposed a many-to-many mapping mechanism to learn different knowledge among different layers.

Moreover, one of the main limitations of these methods is that the student model should share a similar architecture with the teacher model. It is because these methods align the internal representations between teacher and student models and can only be applied to compressing the layer of Transformer. However, our proposed EGKD could perform heterogeneous model compression.

6.2 Explanation Methods

Current explanation methods aim to get the attribution of each token in the input to the model prediction, which can be classified as three classes according to different mechanisms: **Gradient-based** explanation methods get the attribution scores by leveraging the gradient of the model. They compute the gradient of each token and obtain the attribution scores by multiplying the gradient with the token embedding [29, 31, 34]. **Perturbation-based** explanation methods perturb the original inputs by masking some tokens. They can get a series of masks and corresponding output logits. The attribution scores can be computed according to a linear function [19, 26]. **Feature selection** explanation methods aim to find a minimal sufficient subset of the original inputs to ensure these features alone suffice for the same prediction to be reached by the model [4, 17]. Figure 2 also presents an example of the explanation results for these three explanation methods. In this article, we design different variants of EGKD according to these typical explanation methods.

Recent work [23] evaluated the explanation methods by utilizing the attribution scores of the teacher model to guide the attention of the student model. And they find the attention method gets the best performance. Actually, our baseline is stronger than this because matching the attention matrix is a far more strict constraint.

7 CONCLUSION AND FUTURE DIRECTION

In this article, we propose EGKD, which can utilize the explanations of the teacher model to help the student model learn better. Experimental results show EGKD could get promising performance on both in-distribution and out-of-distribution tests. And we also verify the effectiveness of EGKD on knowledge distillation with different model architectures, which sheds light on the universality of EGKD.

Because of the universality of EGKD, we think EGKD could be applied to not only model compression but also more fields where vanilla knowledge distillation is effective, such as incremental learning [42]. And we believe that our proposed EGKD could further improve the performance of vanilla knowledge distillation. The greatest limitation of EGKD is the much training time compared to the vanilla knowledge distillation. Efficiently generating explanations and further improving the

efficiency of EGKD are worth exploring in the future, which could lead to the wider application of EGKD. Moreover, further research is required to explore the integration of EGKD into large language models.

REFERENCES

- [1] Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. 2020. Knowledge distillation from internal representations. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020*. AAAI Press, 7350–7357. <https://aaai.org/ojs/index.php/AAAI/article/view/6229>
- [2] 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'20)*. Association for Computational Linguistics, Online, 3256–3274. <https://doi.org/10.18653/v1/2020.emnlp-main.263>
- [3] Jon Barwise. 1993. Heterogeneous reasoning. In *International Conference on Conceptual Structures*. Springer, 64–74.
- [4] Jasmijn Bastings, Wilker Aziz, and Ivan Titov. 2019. Interpretable neural predictions with differentiable binary variables. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2963–2977. <https://doi.org/10.18653/v1/P19-1284>
- [5] Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth PASCAL recognizing textual entailment challenge. In *TAC*.
- [6] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021). <https://arxiv.org/abs/2108.07258>
- [7] Hanjie Chen and Yangfeng Ji. 2022. Adversarial training for improving model robustness? Look at both prediction and interpretation. *arXiv preprint arXiv:2203.12709* (2022). <https://arxiv.org/abs/2203.12709>
- [8] Zihan Chen, Hongbo Zhang, Xiaoji Zhang, and Leqi Zhao. 2018. Quora question pairs. URL <https://www.kaggle.com/c/quora-question-pairs/> (2018).
- [9] 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- [10] Greg Diamos, Shubho Sengupta, Bryan Catanzaro, Mike Chrzanowski, Adam Coates, Erich Elsen, Jesse H. Engel, Awni Y. Hannun, and Sanjeev Sathesh. 2016. Persistent RNNs: Stashing recurrent weights on-chip. In *Proceedings of the 33rd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19–24, 2016 (JMLR Workshop and Conference Proceedings)*, Maria-Florina Balcan and Kilian Q. Weinberger (Eds.), Vol. 48. JMLR.org, 2024–2033. <http://proceedings.mlr.press/v48/diamos16.html>
- [11] William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP'05)*. <https://aclanthology.org/I05-5002>
- [12] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 968–988.
- [13] Manish Gupta and Puneet Agrawal. 2022. Compression of deep learning models for text: A survey. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 16, 4 (2022), 1–55.
- [14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015). <https://arxiv.org/abs/1503.02531>
- [15] Zhongtao Jiang, Yuanzhe Zhang, Zhao Yang, Jun Zhao, and Kang Liu. 2021. Alignment rationale for natural language inference. 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)*. Association for Computational Linguistics, Online, 5372–5387. <https://doi.org/10.18653/v1/2021.acl-long.417>
- [16] 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*. Association for Computational Linguistics, Online, 4163–4174. <https://doi.org/10.18653/v1/2020.findings-emnlp.372>
- [17] Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Austin, Texas, 107–117. <https://doi.org/10.18653/v1/D16-1011>

- [18] Jianquan Li, Xiaokang Liu, Honghong Zhao, Ruifeng Xu, Min Yang, and Yaohong Jin. 2020. BERT-EMD: Many-to-many layer mapping for BERT compression with Earth Mover’s Distance. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP’20)*. Association for Computational Linguistics, Online, 3009–3018. <https://doi.org/10.18653/v1/2020.emnlp-main.242>
- [19] Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220* (2016). <https://arxiv.org/abs/1612.08220>
- [20] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4–9, 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.), 4765–4774. <https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html>
- [21] Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one?. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8–14, 2019, Vancouver, BC, Canada*, Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett (Eds.), 14014–14024. <https://proceedings.neurips.cc/paper/2019/hash/2c601ad9d2ff9bc8b282670cdd54f69f-Abstract.html>
- [22] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP’14)*. 1532–1543.
- [23] Danish Pruthi, Bhuwan Dhingra, Livio Baldini Soares, Michael Collins, Zachary C. Lipton, Graham Neubig, and William W. Cohen. 2020. Evaluating explanations: How much do explanations from the teacher aid students? *arXiv preprint arXiv:2012.00893* (2020). <https://arxiv.org/abs/2012.00893>
- [24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI Blog* 1, 8 (2019), 9.
- [25] 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*. Association for Computational Linguistics, Austin, Texas, 2383–2392. <https://doi.org/10.18653/v1/D16-1264>
- [26] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. “Why Should I Trust You?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13–17, 2016*, Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi (Eds.). ACM, 1135–1144. <https://doi.org/10.1145/2939672.2939778>
- [27] 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* (2019). <https://arxiv.org/abs/1910.01108>
- [28] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2020. Q-BERT: Hessian based ultra low precision quantization of BERT. In *Proc. of AAAI*.
- [29] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034* (2013). <https://arxiv.org/abs/1312.6034>
- [30] Xuelin Situ, Ingrid Zukerman, Cecile Paris, Sameen Maruf, and Gholamreza Haffari. 2021. Learning to explain: Generating stable explanations fast. 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)*. Association for Computational Linguistics, Online, 5340–5355. <https://doi.org/10.18653/v1/2021.acl-long.415>
- [31] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. 2017. SmoothGrad: Removing noise by adding noise. *arXiv preprint arXiv:1706.03825* (2017). <https://arxiv.org/abs/1706.03825>
- [32] 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*. Association for Computational Linguistics, Seattle, Washington, USA, 1631–1642. <https://aclanthology.org/D13-1170>
- [33] 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’19)*. Association for Computational Linguistics, Hong Kong, China, 4323–4332. <https://doi.org/10.18653/v1/D19-1441>
- [34] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017 (Proceedings of Machine Learning Research)*, Doina Precup and Yee Whye Teh (Eds.), Vol. 70. PMLR, 3319–3328. <http://proceedings.mlr.press/v70/sundararajan17a.html>
- [35] Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. 2019. Distilling task-specific knowledge from BERT into simple neural networks. *arXiv preprint arXiv:1903.12136* (2019). <https://arxiv.org/abs/1903.12136>

- [36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30 (2017).
- [37] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9, 2019*. OpenReview.net. <https://openreview.net/forum?id=rJ4km2R5t7>
- [38] Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020. Structured pruning of large language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP'20)*. Association for Computational Linguistics, Online, 6151–6162. <https://doi.org/10.18653/v1/2020.emnlp-main.496>
- [39] Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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'19)*. 6382–6388.
- [40] Joy Whitenack and Erna Yackel. 2002. Making mathematical arguments in the primary grades: The importance of explaining and justifying ideas. (Principles and Standards). *Teaching Children Mathematics* 8, 9 (2002), 524–528.
- [41] 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)*. Association for Computational Linguistics, New Orleans, Louisiana, 1112–1122. <https://doi.org/10.18653/v1/N18-1101>
- [42] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. 2019. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 374–382.
- [43] Canwen Xu, Wangchunshu Zhou, Tao Ge, Ke Xu, Julian McAuley, and Furu Wei. 2021. Beyond preserved accuracy: Evaluating loyalty and robustness of BERT compression. *arXiv preprint arXiv:2109.03228* (2021). <https://arxiv.org/abs/2109.03228>
- [44] Xi Ye and Greg Durrett. 2021. Can explanations be useful for calibrating black box models? *arXiv preprint arXiv:2110.07586* (2021). <https://arxiv.org/abs/2110.07586>
- [45] Ofir Zafri, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8BERT: Quantized 8Bit BERT. *arXiv preprint arXiv:1910.06188* (2019). <https://arxiv.org/abs/1910.06188>

Received 11 April 2023; accepted 14 December 2023