Learnable Relational Knowledge Distillation For Language Model Compression

Feng Hu[1], Kai Zhang[1(✉)], Ye Liu[1], Meikai Bao[1], Xukai Liu[1], Yanjiang Chen[1], Gang Zhou[2], and Qi Liu[1]

[1] State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China, Hefei, China
[2] Information Engineering University, Zhengzhou, Henan
{fenghufh3, liuyer, baomeikai, chthollylxk, yjchen}@mail.ustc.edu.cn, {kkzhang08, qiliuql}@ustc.edu.cn, gzhougzhou@126.com

Abstract. Knowledge distillation (KD) has emerged as an effective model compression method, drawing significant attention. However, existing methods that employ intermediate representations encounter three primary limitations. Firstly, these methods require the manual design of relational knowledge between intermediate representations. Secondly, their application scenarios are restricted, where the teacher and student models need to have the identical representation dimensions or shared vocabularies. Lastly, these methods require additional time or memory expenditures to enhance performance. To address these issues, we propose ***L**earnable **R**elational **K**nowledge **D**istillation* (**LRKD**). Firstly, LRKD autonomously learns relational knowledge via a dual orthogonal projection without manual design. Secondly, LRKD matches the different representation dimensions through the projection and only leverages mean-pooling to obtain the sequence-level representations for alignment, thereby ignoring the influence of the vocabulary. Lastly, we can deploy LRKD without incurring additional overhead. Specifically, we propose a multi-layer projection to construct more sophisticated relational knowledge. Experimental results demonstrate that LRKD outperforms advanced distillation methods.

Keywords: Knowledge distillation · Model compression.

1 Introduction

Recently, Pre-trained Language Models (PLMs) have demonstrated exceptional performance across various natural language processing tasks [5,13,37,38,39]. However, their large-scale parameters pose challenges for deployment in scenarios with limited computational and storage resources. Consequently, effectively reducing the size of model parameters has become a critical issue in language model deployment. Knowledge Distillation (KD) [10] has emerged as a promising solution to this problem. KD aims to compress these huge PLMs into smaller, more efficient models with minimal performance degradation.

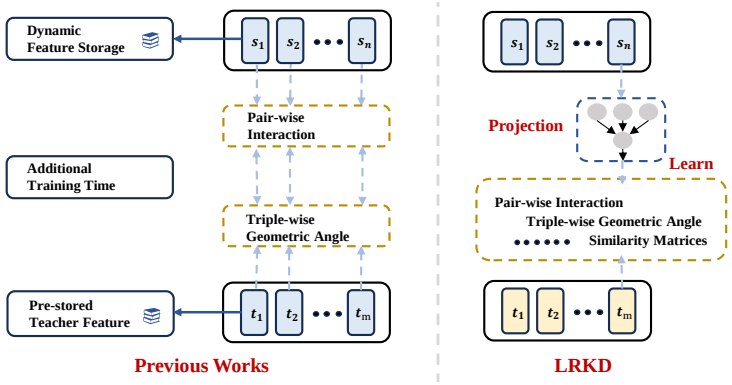


Fig. 1: Comparison of LRKD with previous work. Yellow and blue blocks represent the intermediate representations in the model, and different colors indicate the different architectures that the two models can adopt.

Existing knowledge distillation methods are broadly categorized into three primary approaches: response-based, feature-based, and relation-based methods [7]. Response-based methods focus on directly distilling the final outputs from the teacher model. In contrast, feature-based and relation-based approaches not only align the final outputs but also leverage the intermediate representations for alignment. Specifically, feature-based methods aim to directly align features extracted from the intermediate layers through diverse alignment strategies, like ACKD [9]. For relation-based methods, CKD, MGSKD, and ReAugKD [17,12,36] leverage instance relations or similarity matrices between representations to extract relational knowledge and then align this knowledge.

Although the methods of utilizing representations for alignment are effective, especially for relation-based KD methods, as shown in Fig 1, these methods exhibit three primary limitations. Firstly, these methods necessitate the manual design of relational knowledge, relying on specific scenarios, such as utilizing pair-wise interactions and triple-wise geometric angles to construct sophisticated relational knowledge from several directions or multi-granularity language units [17,12]. However, their manual design for specific scenarios is complex and time-consuming.

Secondly, these methods do not support distillation across different model architectures. Specifically, to calculate the difference between two models' representations, these methods require identical intermediate representation dimensions or shared vocabularies. However, this condition is hardly satisfied for certain language models in practical scenarios, especially for small-sized models, as smaller models with identical representation dimensions or shared vocabularies may not be available.

Finally, previous methods depend on extra representations memory [36,9] or increased training times [34] to enhance the student model's performance. Nevertheless, this demand for extra resources markedly diminishes the efficiency of

the distillation process during both training and inference, especially in environments with limited resources.

To address these challenges, we propose an innovative Learnable Relational Knowledge Distillation (LRKD) framework. Firstly, LRKD leverages a dual orthogonal projection to learn relational knowledge relevant to specific tasks instead of manual design. Secondly, LRKD harmonizes the dimensional disparities between models via the projection and employs mean-pooling to extract sample representations from token representations, thus avoiding the impact of divergent vocabularies. Lastly, LRKD requires no additional computational overhead, relying solely on a multi-layer projection to capture more intricate relational knowledge that is omitted during inference.

In summary, the main contribution of this paper can be summarized as follows:

- We propose a novel knowledge distillation framework. In this framework, we introduce a dual orthogonal projection that autonomously learns relational knowledge during the training phase.
- Our method serves as a universal framework for knowledge distillation, adaptable to various distillation scenarios. We further propose a multi-layer projection to capture more comprehensive relational knowledge.
- We systematically validate LRKD on the GLUE benchmark and show its superior performance over state-of-the-art baselines. We also verify the effectiveness of our method on Large Language Models (LLMs). We will release our codes publicly available at https://github.com/python-bruce/LRKD.

2 Related Work

2.1 Projection-free Knowledge Distillation

Knowledge distillation methods have emerged as a highly promising approach for model compression, attracting significant interest due to their impressive performance. Traditional KD methods are broadly classified into three categories: response-based, feature-based, and relation-based. The response-based KD method, like Vinilla KD [10], utilized the prediction output of the teacher model to provide an additional supervisory signal for the student model. This approach was applied to the BERT model by DistilBERT [22], achieving compression of the teacher model with minimal performance degradation. Nevertheless, these methods don't utilize the intermediate representations to transfer knowledge from the teacher.

For feature-based methods, prior research aimed to directly align intermediate features of both teacher and student models through diverse alignment strategies. FitNet [20] proposed a direct matching of the intermediate representations of the teacher and student. For the BERT model, PKD, TinyBERT, and MobileBERT [24,11,25] further considered the intermediate representations to transfer knowledge from the teacher. CRD [26] adopted contrastive distillation on intermediate representations, which utilized the relation between different

samples. ACKD [9] further integrated contrastive distillation with a SAR strategy to focus on hard samples and utilized dynamic feature storage to increase the number of negative samples. However, directly aligning the representations can't fully transfer knowledge from the teacher model.

For relation-based methods, various strategies have been developed to construct relational knowledge. The work by ReAugKD [36] facilitated the transfer of relational knowledge via similarity matrix distributions and stored the teacher model's intermediate representations and final outputs for retrieval. Meanwhile, CKD [17] utilized pair-wise interactions and triplet-wise geometric angles to construct relational knowledge between token representations, incorporating both horizontal and vertical directions. Further advancements by MGSKD [12], it designed multi-granularity relational knowledge across token-level, span-level, and sample-level representations. However, these methods rely on the manual design of relational knowledge.

Recently, an innovative attribution-driven knowledge distillation framework was proposed by AD-KD [34], which efficiently transfers attribution knowledge from teacher to student. Nevertheless, this approach requires the acquisition of attribution knowledge, necessitating two backpropagations for each parameter update of the student model. Different from these methods, our LRKD framework autonomously learns relational knowledge through a dual projection and can be applied to different distillation scenarios without additional assumptions.

2.2 Projection-dependent Knowledge Distillation

Several studies have also investigated the use of projection techniques in knowledge distillation. TinyBERT [11] firstly proposed matching the different representation dimensions of the student and teacher through projection. In the realm of computer vision, Ensemble [3] expanded on this concept by proposing an ensemble of projections to enhance KD performance. SRD [16] suggested a straightforward KD method utilizing a linear projection coupled with appropriate normalization, resulting in significant performance enhancements. Furthermore, V_kD [15] advocated for the use of orthogonal projection over linear projection to maintain intra-batch feature similarity. Different from these approaches, we present a novel knowledge distillation method that leverages a dual orthogonal projection consisting of multiple layers to capture more sophisticated relational knowledge.

3 Method

In this section, we first introduce the relevant definitions of knowledge distillation and then present our LRKD framework, which is shown in Fig 2. There are four different components of loss functions in the student model's training process: cross-entropy loss, knowledge distillation loss, and dual projection loss (DPL), which is made up of the student projection loss (SPL) and the teacher projection loss (TPL).

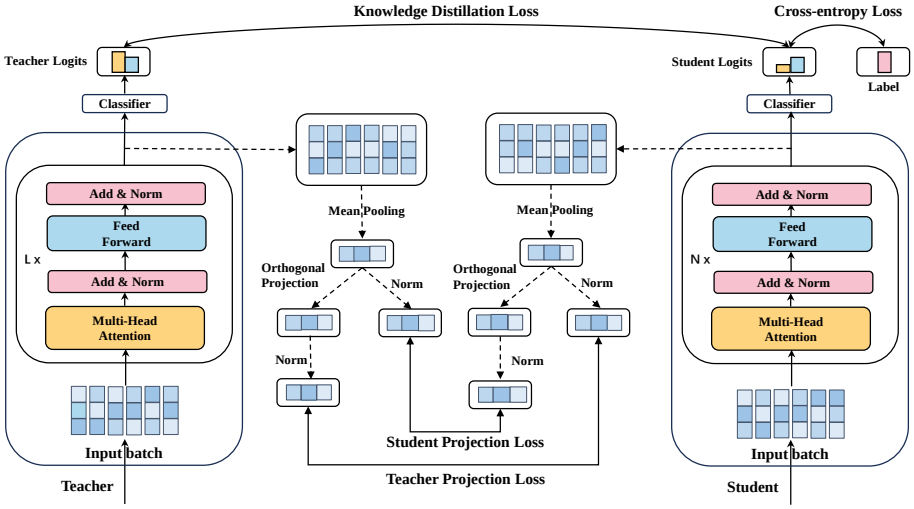


Fig. 2: The overall architecture of our LRKD framework for BERT compression.

3.1 Preliminary

Projection-free Knowledge Distillation. Given an input text, we denote the input sequence as $x = [x_1, x_2, \ldots, x_n]$, where n is the sequence length and each x_i represents one token. The tokens are converted to the d-dimensional embedding sequence $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n] \in R^{n \times d}$ through the embedding layer. For the sake of clarity, we treat the embedding layer as the 0-th layer and set $\boldsymbol{H}^0 = \boldsymbol{E}$. Subsequently, these embeddings $\boldsymbol{H}^0$ are sequentially processed through L stacked transformer layers. In the l-th layer, the output representations $\boldsymbol{H}^{l-1}$ from the preceding layer serve as the input. This layer employs multi-head attention (MHA) and a position-wise feed-forward network (FFN) to refine these representations, yielding the updated $\boldsymbol{H}^l$. Finally, a task-specific head is applied to $\boldsymbol{H}^L$ to get the final output $\boldsymbol{Z}$.

For the traditional KD method, the student will be trained by using the loss function as follows:

$$\begin{aligned} \mathcal{L} &= \alpha \mathcal{L}_{CE} + (1-\alpha)\mathcal{L}_{KD} \\ &= \alpha \cdot \mathrm{CE}(\sigma(\boldsymbol{Z}_s), \boldsymbol{Y}) \\ &+ (1-\alpha) \cdot \mathrm{KL}(\sigma(\boldsymbol{Z}_t/\tau) \| \sigma(\boldsymbol{Z}_s/\tau)). \end{aligned} \tag{1}$$

The task-specific loss, denoted as $\mathcal{L}_{CE}$ typically employs the cross-entropy loss function $\mathrm{CE}(\cdot,\cdot)$. The distillation loss, represented by $\mathcal{L}_{KD}$ commonly leverages the Kullback-Leibler divergence $\mathrm{KL}(\cdot,\cdot)$ to quantify the discrepancy in the predicted probability distributions between a student model and its teacher model. The hyperparameter α modulates the loss contributions. And the parameter τ denotes the distillation temperature. The softmax function is denoted

by σ, and $\boldsymbol{Y}$ represents the ground-truth label. The output logits for a batch from the teacher and student models are represented by $\boldsymbol{Z}_t$ and $\boldsymbol{Z}_s$, respectively.

Projection-dependent Knowledge Distillation. In the field of computer vision, there are some projection-dependent knowledge distillation methods. Taking V_kD [15] as an example, the loss function can be formulated as follows:

$$\mathcal{L} = \frac{1}{2}||\boldsymbol{H}_s\boldsymbol{W} - standard(\boldsymbol{H}_t)||_2^2, \tag{2}$$

where $\boldsymbol{H}_s$ and $\boldsymbol{H}_t$ are the representations of the student model and the teacher model before the classifier, respectively. $standard(\cdot)$ denotes the standardisation and whitening operation, and $\boldsymbol{H}_t$ through $standard(\cdot)$ satisfies $(\boldsymbol{H}_t)^T\boldsymbol{H}_t = \boldsymbol{I}$. $\boldsymbol{W}$ is an orthogonal projection. Distinct from previous studies, our approach involves detailed optimizations of student and teacher models within transformer structures for distillation scenarios. And by introducing dual projection and multi-layer projection, we construct more effective relational knowledge to fully transfer knowledge from the teacher. Based on these, our architecture can be applied to different distillation scenarios.

3.2 Feature Transformation

Current KD methods utilizing intermediate representations require manual design of relational knowledge and then alignment. To avoid the manual design, we introduce a projection to transform the intermediate representations prior to their alignment. This projection is trained to capture relational knowledge between representations, thereby enabling the model to more comprehensively inherit knowledge from the teacher model.

Linear Projection. Specifically, we utilize the last-layer intermediate representations of the two models to transfer the knowledge. Following previous works, we employ the method described in MGSKD [12], which involves mean-pooling all the token representations within a text sample to generate a comprehensive, sample-level representation:

$$\tilde{\boldsymbol{h}} = \text{Pool}(\boldsymbol{H}^L), \tag{3}$$

and we simply gather all the sample representations in a mini-batch, which denotes $\boldsymbol{H}$:

$$\boldsymbol{H} = [\tilde{\boldsymbol{h}}_1, \tilde{\boldsymbol{h}}_2, \ldots, \tilde{\boldsymbol{h}}_b], \tag{4}$$

which b denotes the batch size. We derive the sequence-level representations $\boldsymbol{H}_s$ and $\boldsymbol{H}_t$, where the superscript s and t denote the student model and the teacher model, respectively. Then, we apply a linear projection $\boldsymbol{W}_s$ to transform the representations $\boldsymbol{H}_s$, as exemplified by the student representations:

$$\boldsymbol{H}_s^f = \boldsymbol{H}_s\boldsymbol{W}_s. \tag{5}$$

Orthogonal Projection. The linear projection transforms representations before alignment, and this projection is discarded after training. However, when the projection is too complex, it can disrupt the original relations between the representations after transformation, which leads to inconsistencies between the relations obtained after transformation and the original relations between representations. To solve this problem, inspired by works [18,15], we find that orthogonal transformation can keep the pair-wise angles of two features, which means that the relations between representations can be mostly maintained after orthogonal transformation. So we utilize an orthogonal projection instead of a linear projection in our LRKD framework.

To ensure the orthogonality of the projection throughout the distillation process, we follow the previous work [18] to employ Cayley parameterization. Specifically, we construct the orthogonal matrix using $\boldsymbol{W}_s = (\boldsymbol{I} + \boldsymbol{Q})(\boldsymbol{I} - \boldsymbol{Q})^{-1}$, where $\mathbf{Q}$ is a skew-symmetric matrix satisfying $\boldsymbol{Q} = -\boldsymbol{Q}^\top$ and $\boldsymbol{I}$ is an identity matrix.

Multi-layer Projection. To explore relational knowledge across different complexities, we introduce a multi-layer projection (MLP) [21]. Given two orthogonal matrices, $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, their product $\boldsymbol{W} = \boldsymbol{W}_1\boldsymbol{W}_2$ remains orthogonal. Leveraging this property, we reformulate Eq. (5) as follows:

$$\boldsymbol{H}_s^f = \boldsymbol{H}_s \boldsymbol{P}_s^m, \tag{6}$$

here, $\boldsymbol{P}_s^m$ represents the student projection consisting of m orthogonal matrices. By setting different numbers of layers m for tasks, we can more effectively transfer knowledge from the teacher model.

3.3 Dual Projection Loss

Our dual projection loss (DPL) consists of the student projection loss (SPL) and the teacher projection loss (TPL). We first elaborate on the student projection loss (SPL).

Student Projection Loss. To transfer knowledge from the teacher model to the student model, a direct strategy is to minimize the difference between their representations, such as L2 distance (MSE). However, there may be a magnitude gap between the representations in teacher and student models at the early phase of distillation since the projection has just been initialized and is poorly trained. Under this circumstance, the student is likely to fall into a local optimum. To enable smooth knowledge distillation, we normalize the representations before minimizing the difference. So we define the student projection loss (SPL) as follows:

$$\mathcal{L}_{SPL} = \frac{1}{2}||norm(\boldsymbol{H}_s^f) - norm(\boldsymbol{H}_t)||_2^2, \tag{7}$$

where $\boldsymbol{H}_s^f$ originates from Eq. (6) and $norm(\cdot)$ denotes that LayerNorm [1] is performed on the representation dimension for the $\boldsymbol{H}_s^f$ and $\boldsymbol{H}_t$.

Teacher Projection Loss. To more fully transfer the knowledge, we also introduce another projection to transform the teacher's representations and then align the student's representations. Similarly, we obtain the teacher projection loss (TPL) as follows:

$$\boldsymbol{H}_t^f = \boldsymbol{H}_t \boldsymbol{P}_t^m, \tag{8}$$

$$\mathcal{L}_{TPL} = \frac{1}{2}||norm(\boldsymbol{H}_t^f) - norm(\boldsymbol{H}_s)||_2^2, \tag{9}$$

where $\boldsymbol{P}_t^m$ represents teacher projection consisting of m orthogonal matrices.

Combining the two loss terms above, we obtain our dual projection loss (DPL) as follows:

$$\mathcal{L}_{DPL} = \gamma \mathcal{L}_{SPL} + (1-\gamma)\mathcal{L}_{TPL}, \tag{10}$$

where γ adjusts the scales of two losses. The dual projection is integrated into the student model during the training phase and subsequently removed for inference. As a result, the trained student model performs inference without any additional computational overhead.

The projection learns relational knowledge. To more intuitively demonstrate the role of our projection in distillation, we examine the loss term $\mathcal{L}_{SPL}$. For simplicity, we omit the normalization step. The loss term is expressed as follows:

$$\mathcal{L}_{SPL} = \frac{1}{2}||\boldsymbol{H}_s \boldsymbol{P}_s - \boldsymbol{H}_t||_2^2. \tag{11}$$

Taking the derivative with respect to $\boldsymbol{P}_s$, we obtain the fixed-point solution for the projection as follows:

$$\boldsymbol{H}_s^T \boldsymbol{H}_s \boldsymbol{P}_s = \boldsymbol{H}_s^T \boldsymbol{H}_t, \tag{12}$$

the terms $\boldsymbol{H}_s^T \boldsymbol{H}_s$ and $\boldsymbol{H}_s^T \boldsymbol{H}_t$ represent self-correlation and cross-correlation matrices, respectively. The projection $\boldsymbol{P}_s$ learns the relations between these two correlation matrices, effectively mapping between these correlation spaces. This indicates that the projection weights encode relations from previous samples and then construct relational knowledge for representations during the training phase. It is demonstrated that explicit construction of relational knowledge is not necessary. Instead, these relations can be effectively learned through the projection.

3.4 Overall Objective

In line with previous works, we combine our loss term $\mathcal{L}_{DPL}$ with the original cross-entropy loss $\mathcal{L}_{CE}$ and the distillation loss $\mathcal{L}_{KD}$. Thus, the overall training objective is defined as follows:

$$\mathcal{L} = \alpha \mathcal{L}_{CE} + (1-\alpha)\mathcal{L}_{KD} + \beta \mathcal{L}_{DPL}. \tag{13}$$

$\mathcal{L}_{DPL}$ comes from Eq. (10) and the hyperparameter β controls the trade-off of this term.

4 Experiments

Datesets. Following the previous works, we conducted our experiments on eight tasks of the GLUE benchmark [29], including CoLA [32], SST-2 [23], MRPC [6], STS-B [2], QQP [4], MNLI [33], QNLI [19], and RTE. We report accuracy metrics for MNLI, SST-2, QNLI, MRPC, QQP, and RTE. For MRPC, we additionally provide the F1 score. Performance on CoLA is assessed using the Matthews correlation coefficient, while STS-B is evaluated through Spearman's rank correlation coefficient.

Baseline Methods. We compare LRKD against several state-of-the-art feature-based and relation-based KD methods, in which these methods utilize hidden representations to transfer knowledge. The traditional KD method used for comparison is Vanilla KD [10]. The feature-based KD methods aim to directly align intermediate features of both teacher and student models through diverse alignment strategies, like PKD [24], TinyBERT [11] and ACKD [9]. The relation-based methods leverage instance relations, such as pair-wise interactions and triple-wise geometric angles, or similarity matrices between representations to extract relational knowledge and then align this knowledge, like CKD [17], MGSKD [12] and ReAugKD[36]. MetaDistill [41] first introduced meta-learning into knowledge distillation to obtain student's feedback. Recently, AD-KD [34] proposed an attribution-driven approach to transfer knowledge.

Our LRKD method is a task-specific distillation method, but for MiniLM [31,30], these methods are task-agnostic distillation methods. The setting is different from ours, so we don't select these methods as the baseline methods. However, the task-specific distillation methods can combine with the task-agnostic distillation methods to further improve performance.

Implementation Details. We conduct our experiments based on the Pytorch platform. Following previous work [17], we choose BERT_{base} as our teacher model and utilize a smaller BERT released by work [28] with 6 Transformer layers, 768 hidden neurons and 12 attention heads as our student model. We first fine-tune the pre-trained BERT_{base} in the downstream tasks. The maximum sequence length is 128 and the AdamW [14] optimizer is adopted. We choose the initial learning rate and batch size from {2e-5, 3e-5, 5e-5} and {8, 16, 32}, respectively. Then, we implement LRKD to distill the student model. We search for the optimal learning rate in {2e-5, 3e-5, 4e-5, 5e-5}, batch size in {16, 32}, α in {0.7, 0.8, 0.9, 1.0} and temperature in {1.0, 2.0, 3.0, 4.0}. And for the hyperparameter β, we choose from {1.0, 10.0, 15.0}. For the hyperparameter γ, we choose around 0.3. We choose a two-layer projection for all the downstream tasks. Though our method introduces a dual orthogonal projection that increases additional parameters during the training stage, this part of the parameters is approximately one percent of the model parameters. And during the inference process, this projection will be removed, so it does not increase any parameters for the student.

5 Results and Analysis

5.1 Main Results

Table 1: The experimental results of LRKD and other previous works on the GLUE benchmark. The best result is in **bold**, while the second-best result is underlined.

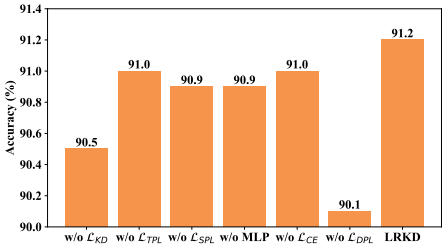
Model	Params	CoLA (Mcc)	MNLI-(m/mm) (Acc)	SST-2 (Acc)	QNLI (Acc)	MRPC (F1)	QQP (Acc)	RTE (Acc)	STS-B (Spear)	Avg
$BERT_{base}$(Teacher)	110M	60.4	84.7/84.5	93.3	91.7	91.6	91.4	71.4	89.6	84.3
$BERT_6$(Student)	66M	51.2	81.7/82.6	91.0	89.3	89.2	90.4	66.1	88.3	80.9
Vanilla KD [10]	66M	53.6	82.7/83.1	91.1	90.1	89.4	90.5	66.8	88.7	81.6
TinyBERT [11]	66M	53.8	83.1/83.4	92.3	89.9	88.8	90.5	66.9	88.3	81.7
PKD [24]	66M	45.5	81.3/-	91.3	88.4	85.7	88.4	66.5	86.2	79.2
CKD [17]	66M	55.1	**83.6**/84.1	**93.0**	90.5	89.6	91.2	67.3	89.0	82.4
MGSKD [12]	66M	49.1	83.3/83.9	91.7	90.3	89.8	91.2	67.9	88.5	81.5
MetaDistill [41]	66M	58.6	83.5/83.8	92.3	90.4	91.1	91.0	69.4	89.1	83.2
ReAugKD [36]	66M	59.4	-/-	92.5	90.7	-	91.2	70.4	-	81.8
ACKD [9]	66M	**59.7**	**83.6**/83.9	92.3	90.6	91.0	**91.3**	69.7	89.1	83.4
AD-KD [34]	66M	58.3	83.4/84.2	91.9	**91.2**	91.2	91.2	**70.9**	**89.2**	83.4
LRKD	66M	59.4	83.5/**84.3**	92.3	**91.2**	**91.4**	**91.3**	**70.9**	**89.2**	**83.7**

The results are presented in Table 1. LRKD outperforms other baseline methods on the majority of the dataset. The table demonstrates that some feature-based and relation-based KD methods even underperform Vanilla KD in some datasets, such as MGSKD in CoLA and PKD in several datasets. In comparison, our LRKD framework outperforms all relation-based KD methods, such as MGSKD, CKD, and ReAugKD. More concretely, LRKD yields an average improvement of 1.3, 1.9, and 2.2 points over CKD, ReAugKD, and MGSKD, respectively. This means that relational knowledge learned through dual projection is more effective than that manually constructed.

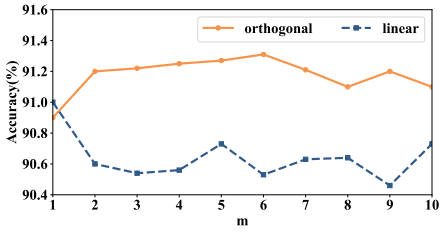
On the other hand, certain methods, such as ACKD and ReAugKD, necessitate additional memory to store model representations during the training or inference stages. In contrast, LRKD operates without such memory demand. Although AD-KD's results are only slightly inferior to LRKD's results, its training process requires two backpropagation calculations, which increases its training time. As shown in Table 2, AD-KD's training time is approximately 1.6 times that of our method.

Table 2: The results of training time compared to AD-KD on several larger datasets.

Datasets	MNLI	QNLI	QQP	SST-2
AD-KD	2.5hr	50min	4hr	32min
LRKD	1.5hr	39min	2.5hr	20min



(a) Ablation study of different components.



(b) Ablation study of orthogonal or linear projection.

Fig. 3: Ablation study on the QNLI development set.

5.2 Ablation Studies

Impact of Loss Terms. As mentioned in the multi-layer projection and dual projection loss sections, we proposed the overall loss and the multi-layer projection (MLP) structure. We assess the impact of these components on the QNLI development set. Specifically, these components include cross-entropy loss L_{CD}, knowledge distillation loss L_{KD}, dual projection loss L_{DPL} that consists of student projection loss L_{SPL} and teacher projection loss L_{TPL}, as well as the MLP structure. As shown in Fig 3a, removing any single loss term results in a decline in overall performance. The removal of either L_{SPL} or L_{TPL} within L_{DPL} leads to a decline in student model performance, indicating that both components are essential in L_{DPL} for transferring comprehensive knowledge from the teacher model. Furthermore, the lack of an MLP in our dual projection framework results in decreased performance, as it facilitates the learning of more complex relational knowledge.

Impact of Different Projections. As mentioned in the orthogonal projection section, we proposed an orthogonal projection to transform the intermediate representations. We conduct the experiment to investigate the performance of the student model when adopting different types of projections, such as orthogonal projection or linear projection. As illustrated in Fig 3b, in nearly all cases except $m = 1$, orthogonal projection significantly outperforms linear projection. Moreover, as the number of layers increases, the gap between orthogonal and linear projections widens. This means that an orthogonal projection can preserve relations between the representations. And as mentioned in the multi-layer projection section, increasing the number of layers enhances the complexity of our projection, thereby constructing more complex relational knowledge to enhance the student model.

5.3 Discussion and Analysis

Projection of Different Layers. As mentioned in the multi-layer projection section, we proposed a multi-layer projection to construct relational knowledge

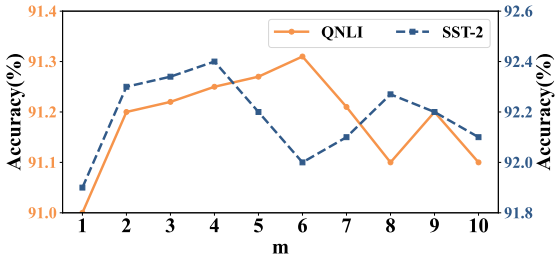


Fig. 4: The results of the orthogonal projection of different layers on the QNLI and SST-2 development sets.

across different complexities. To further explore the impact of different numbers of layers on various datasets, we conduct experiments with different values of m on the QNLI (108K) and SST-2 (67K) development sets. We find that the optimal number of layers varies across different datasets. Specifically, the optimal number of layers increases with larger datasets, indicating that the value m represents relational knowledge of different complexity and that more complex relational knowledge could be learned by selecting a value of m.

Different Distillation Scenarios. Traditional KD methods necessitate similar structures between student and teacher models. However, our approach diverges from this requirement. To verify the generalization of our method across different frameworks, we set up two distinct distillation scenarios, and currently, almost all existing methods cannot adapt to both. From Table 3a, we can observe that our method remains effective with different vocabularies, which many traditional methods cannot achieve. Additionally, Table 3b shows that our method achieves state-of-the-art performance even compared with AD-KD, where the dimensions of the representations between the two models are inconsistent. These two settings further demonstrate the versatility of our approach.

Table 3: The results of LRKD for different teacher and student models.

(a) The results of LRKD when the teacher model is RoBERTa$_{base}$ and the student model is BERT$_{base}$.

Model	**QNLI (Acc)**	**SST-2 (Acc)**
RoBERTa$_{base}$(Teacher)	92.8	94.8
BERT$_{base}$(Student)	91.7	93.3
Vanilla KD	92.2	93.6
LRKD	**92.5**	**94.2**

(b) The results of LRKD when the teacher model is BERT$_{large}$ and the student model is BERT$_6$.

Model	**QNLI (Acc)**	**SST-2 (Acc)**
BERT$_{large}$(Teacher)	92.6	93.8
BERT$_6$(Student)	89.3	91.0
Vanilla KD	90.0	91.3
AD-KD	90.6	91.7
LRKD	**91.0**	**92.5**

The performance on the LLMs. We conduct an evaluation of our LRKD framework using several instruction-following datasets. Specifically, we utilize the databricksdolly-15k dataset, processed by MiniLLM [8], comprising approximately 11k samples for training, 1k for validation, and 500 for testing. Additionally, we incorporate Self-Instruct (SelfInst), Vicuna-Evaluation (VicunaEval), Super Natural Instructions (S-NI), and Unnatural Instructions (UnNI) as supplementary test sets to ensure a comprehensive assessment. For our experiments, TinyLLaMA-1.1B [40] is selected as the student language model, while LLaMA2-7B [27] serves as the teacher model. We fine-tune the student and the teacher with LoRA. We combine our approach with baseline methods such as KL [10], which employs standard Kullback-Leibler divergence; RKL, which reverses the distribution order in KL divergence; and AKL [35], which integrates an adaptive fusion of KL and RKL.

Our LRKD framework can be integrated into these baseline methods, and the results of the performance are presented in Table 4. The enhancements observed in student performance underscore the effectiveness of our approach and the applicability of LRKD to synergize with other logit-based methods in Large Language Models (LLMs).

Table 4: The experimental results of LRKD framework in LLMs. "+LRKD" denotes that employs our framework into these baselines.

Model	**Dolly**	**SelfInst**	**VicunaEval**	**S-NI**	**UnNI**	**Avg.**
Teacher	28.4	21.2	18.7	32.4	32.5	26.7
SFT	23.1	15.0	16.5	27.9	26.3	21.8
KL	25.6	17.1	16.5	29.4	29.4	23.6
KL+LRKD	**27.0**	**18.2**	**18.9**	**32.2**	**31.7**	**25.6**
RKL	24.7	17.4	16.9	29.7	29.5	23.7
RKL+LRKD	**25.9**	**19.3**	**17.9**	**32.9**	**33.7**	**25.9**
AKL	24.9	17.0	16.8	29.2	28.9	23.4
AKL+LRKD	**27.1**	**18.6**	**19.1**	**32.5**	**32.5**	**26.0**

6 Conclusion

In this paper, we introduce an innovative knowledge distillation framework. Firstly, the projection autonomously learns relational knowledge instead of manual design through a dual orthogonal projection. Secondly, LRKD uses mean-pooling to obtain sequence-level representations based on token representations and can match the different dimensions of the two models through projection. So, we can apply LRKD across different distillation scenarios. Finally, our method obviates the need for additional memory or training time assumptions. Extensive experiments, including ablation studies, demonstrate the effectiveness of LRKD and the proposed loss function.

7 Acknowledgment

This research was partially supported by grants from the National Natural Science Foundation of China (Grants No. 62337001,62406303), the Key Technologies R & D Program of Anhui Province (No. 202423k09020039), the Anhui Provincial Natural Science Foundation (No. 2308085QF229), Anhui Science and Technology Innovation Plan (No.202423k09020010) and the Fundamental Research Funds for the Central Universities (No. WK2150110034).

References

1. Ba, J., Kiros, J.R., Hinton, G.E.: Layer normalization. ArXiv **abs/1607.06450** (2016), https://api.semanticscholar.org/CorpusID:8236317
2. Cer, D.M., Diab, M.T., Agirre, E., Lopez-Gazpio, I., Specia, L.: Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In: International Workshop on Semantic Evaluation (2017), https://api.semanticscholar.org/CorpusID:4421747
3. Chen, Y., Wang, S., Liu, J., Xu, X., de Hoog, F., Huang, Z.: Improved feature distillation via projector ensemble. ArXiv **abs/2210.15274** (2022), https://api.semanticscholar.org/CorpusID:253157785
4. Chen, Z., Zhang, H., Zhang, X., Zhao, L.: Quora question pairs (2017), https://api.semanticscholar.org/CorpusID:233225749
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: North American Chapter of the Association for Computational Linguistics (2019), https://api.semanticscholar.org/CorpusID:52967399
6. Dolan, W.B., Brockett, C.: Automatically constructing a corpus of sentential paraphrases. In: International Joint Conference on Natural Language Processing (2005), https://api.semanticscholar.org/CorpusID:16639476
7. Gou, J., Yu, B., Maybank, S.J., Tao, D.: Knowledge distillation: A survey. International Journal of Computer Vision **129**, 1789 – 1819 (2020), https://api.semanticscholar.org/CorpusID:219559263
8. Gu, Y., Dong, L., Wei, F., Huang, M.: Knowledge distillation of large language models. ArXiv **abs/2306.08543** (2023), https://api.semanticscholar.org/CorpusID:259164722
9. Guo, J., Liu, J., Wang, Z., Ma, Y., Gong, R., Xu, K., Liu, X.: Adaptive contrastive knowledge distillation for bert compression. In: Annual Meeting of the Association for Computational Linguistics (2023), https://api.semanticscholar.org/CorpusID:259858751
10. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. ArXiv **abs/1503.02531** (2015), https://api.semanticscholar.org/CorpusID:7200347
11. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., Liu, Q.: Tinybert: Distilling bert for natural language understanding. ArXiv **abs/1909.10351** (2019), https://api.semanticscholar.org/CorpusID:202719327
12. Liu, C., Tao, C., Feng, J., Zhao, D.: Multi-granularity structural knowledge distillation for language model compression. In: Annual Meeting of the Association for Computational Linguistics (2022), https://api.semanticscholar.org/CorpusID:248780060

13. Liu, Y., Zhang, K., Huang, Z., Wang, K., Zhang, Y., Liu, Q., Chen, E.: Enhancing hierarchical text classification through knowledge graph integration. In: Findings of the Association for Computational Linguistics: ACL 2023. pp. 5797–5810 (2023)
14. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2017), https://api.semanticscholar.org/CorpusID:53592270
15. Miles, R., Elezi, I., Deng, J.: Vkd: Improving knowledge distillation using orthogonal projections. ArXiv **abs/2403.06213** (2024), https://api.semanticscholar.org/CorpusID:268358422
16. Miles, R., Mikolajczyk, K.: Understanding the role of the projector in knowledge distillation. In: AAAI Conference on Artificial Intelligence (2023), https://api.semanticscholar.org/CorpusID:257632008
17. Park, G., Kim, G., Yang, E.: Distilling linguistic context for language model compression. In: Conference on Empirical Methods in Natural Language Processing (2021), https://api.semanticscholar.org/CorpusID:237563200
18. Qiu, Z., yu Liu, W., Feng, H., Xue, Y., Feng, Y., Liu, Z., Zhang, D., Weller, A., Scholkopf, B.: Controlling text-to-image diffusion by orthogonal finetuning. ArXiv **abs/2306.07280** (2023), https://api.semanticscholar.org/CorpusID:259138650
19. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. In: Conference on Empirical Methods in Natural Language Processing (2016), https://api.semanticscholar.org/CorpusID:11816014
20. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. CoRR **abs/1412.6550** (2014), https://api.semanticscholar.org/CorpusID:2723173
21. Rosenblatt, F.: The perceptron: a probabilistic model for information storage and organization in the brain. Psychological review **65 6**, 386–408 (1958), https://api.semanticscholar.org/CorpusID:12781225
22. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. ArXiv **abs/1910.01108** (2019), https://api.semanticscholar.org/CorpusID:203626972
23. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C.D., Ng, A., Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Conference on Empirical Methods in Natural Language Processing (2013), https://api.semanticscholar.org/CorpusID:990233
24. Sun, S., Cheng, Y., Gan, Z., Liu, J.: Patient knowledge distillation for bert model compression. In: Conference on Empirical Methods in Natural Language Processing (2019), https://api.semanticscholar.org/CorpusID:201670719
25. Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., Zhou, D.: Mobilebert: a compact task-agnostic bert for resource-limited devices. In: Annual Meeting of the Association for Computational Linguistics (2020), https://api.semanticscholar.org/CorpusID:215238853
26. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. ArXiv **abs/1910.10699** (2019), https://api.semanticscholar.org/CorpusID:204838340
27. Touvron, H., Martin, L., Stone, K.R.: Llama 2: Open foundation and fine-tuned chat models. ArXiv **abs/2307.09288** (2023), https://api.semanticscholar.org/CorpusID:259950998

28. Turc, I., Chang, M.W., Lee, K., Toutanova, K.: Well-read students learn better: On the importance of pre-training compact models. arXiv: Computation and Language (2019), https://api.semanticscholar.org/CorpusID:202889175
29. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: Glue: A multi-task benchmark and analysis platform for natural language understanding. In: BlackboxNLP@EMNLP (2018), https://api.semanticscholar.org/CorpusID:5034059
30. Wang, W., Bao, H., Huang, S., Dong, L., Wei, F.: Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. ArXiv **abs/2012.15828** (2020), https://api.semanticscholar.org/CorpusID:229923069
31. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. ArXiv **abs/2002.10957** (2020), https://api.semanticscholar.org/CorpusID:211296536
32. Warstadt, A., Singh, A., Bowman, S.R.: Neural network acceptability judgments. Transactions of the Association for Computational Linguistics **7**, 625–641 (2018), https://api.semanticscholar.org/CorpusID:44072099
33. Williams, A., Nangia, N., Bowman, S.R.: A broad-coverage challenge corpus for sentence understanding through inference. In: North American Chapter of the Association for Computational Linguistics (2017), https://api.semanticscholar.org/CorpusID:3432876
34. Wu, S., Chen, H., Quan, X., Wang, Q., Wang, R.: Ad-kd: Attribution-driven knowledge distillation for language model compression. In: Annual Meeting of the Association for Computational Linguistics (2023), https://api.semanticscholar.org/CorpusID:258740796
35. Wu, T., Tao, C., Wang, J., Zhao, Z., Wong, N.: Rethinking kullback-leibler divergence in knowledge distillation for large language models. ArXiv **abs/2404.02657** (2024), https://api.semanticscholar.org/CorpusID:268876464
36. Zhang, J., Muhamed, A., Anantharaman, A., Wang, G., Chen, C., Zhong, K., Cui, Q., Xu, Y., Zeng, B., Chilimbi, T.M., Chen, Y.: Reaugkd: Retrieval-augmented knowledge distillation for pre-trained language models. In: Annual Meeting of the Association for Computational Linguistics (2023), https://api.semanticscholar.org/CorpusID:259370551
37. Zhang, K., Liu, Q., Qian, H., Xiang, B., Cui, Q., Zhou, J., Chen, E.: Eatn: An efficient adaptive transfer network for aspect-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering **35**(1), 377–389 (2021)
38. Zhang, K., Zhang, H., Liu, Q., Zhao, H., Zhu, H., Chen, E.: Interactive attention transfer network for cross-domain sentiment classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 5773–5780 (2019)
39. Zhang, K., Zhang, K., Zhang, M., Zhao, H., Liu, Q., Wu, W., Chen, E.: Incorporating dynamic semantics into pre-trained language model for aspect-based sentiment analysis. arXiv preprint arXiv:2203.16369 (2022)
40. Zhang, P., Zeng, G., Wang, T., Lu, W.: Tinyllama: An open-source small language model. ArXiv **abs/2401.02385** (2024), https://api.semanticscholar.org/CorpusID:266755802
41. Zhou, W., Xu, C., McAuley, J.: Bert learns to teach: Knowledge distillation with meta learning. In: Annual Meeting of the Association for Computational Linguistics (2021), https://api.semanticscholar.org/CorpusID:237250417