2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Adaptive Knowledge Distillation with Dynamic Weight Allocation

Shaokang Zhang[1,2]*, Yixin Zhang[1], Ning Ran[3] and Liang Wang[1,2]

[1] School of Cyber Security and Computer, Hebei University, Baoding 071000, Hebei, China
[2] Key Laboratory on High Trusted Information System in Hebei Province, Hebei University, Baoding 071000, Hebei, China
[3] College of Electronic Information Engineering, Hebei University, Baoding 071000, Hebei, China
15530204649@163.com

Abstract. Knowledge distillation has become a key technique for compressing pre-trained language models. However, existing methods suffer from some limitations. First, the student model can only imitate the teacher model, but the teacher cannot adapt to the ability of the student model. Second, the student model should focus on learning the knowledge that it is unfamiliar with. Existing methods that distill all the knowledge of the teacher model may bring redundant information. To address these issues, we propose Dynamic Weighted Adaptive Knowledge Distillation, which can adaptively update the teacher model and weight distillation. Specifically, the teacher model is updated according to feedback on the performance of the distilled student model in the independent quiz dataset. We introduce a dynamic weight assignment mechanism that controls the knowledge learned by the student model based on the difference between the teacher model and the student model. Experimental results show that our method outperforms several state-of-the-art methods on multiple datasets.

Keywords: Knowledge Distillation, Pre-trained Language Models, Adaptive Weight Distillation.

1 Introduction

The field of Natural Language Processing (NLP) has made significant progress in recent years and the emergence of pre-trained language models (PLMs) is one of the key factors driving this progress, such as BERT [1] and RoBERTa [2]. They learn rich linguistic knowledge by pre-training a large amount of textual data and can achieve excellent performance in a large number of NLP tasks. However the complexity and computational resource requirements of pre-trained models also pose significant challenges. As a result, many model compression techniques have emerged. Among them, Knowledge Distillation (KD) is an effective PLMs compression technique designed to extract knowledge from larger teacher models to smaller student models with comparable performance.

*Corresponding Author

Existing knowledge distillation methods can be divided into three categories: response-based [3], feature-based [4], and relation-based [5]. Response-based methods directly distill the final output from the output layer of the teacher model, while feature-based and relation-based methods perform distillation by minimizing the difference in the aligned features of the intermediate layers of the teacher and student models. Existing methods still have some limitations: First, in most methods, the student model simply imitates the teacher model. Even though the recent method [6] introduces student perceptual refinement through joint training, the teacher model still cannot be updated according to the capabilities of the student model. Second, recent research work AD-KD [7] combines attribution information to enhance soft-labeled knowledge distillation. However, all the knowledge of the teacher model is averagely refined, which makes the student model unable to focus on learning unfamiliar knowledge, resulting in information redundancy. Moreover, the above methods ignore the high-level attention knowledge in the teacher model.

In this paper, we propose a distillation framework that can adaptively update teacher models and weight distillation. First, we use the traditional distillation method to update the student copy obtained by copying the student model parameters. Then, we feed back its evaluation results on an independent test set to the teacher model through gradient descent, thereby achieving dynamic updating of the teacher model. Second, the adaptive dynamic weight assignment mechanism compares the student model with the teacher model and quantifies the difference indicators. It dynamically readjusts the distillation weights of the differences and the same parts, enabling the student to focus on unfamiliar knowledge. We add a final layer of the attention matrix distillation module to enrich the knowledge contained in the soft labels.

In summary, the main contributions of our work are summarized as follows:

- We introduce a dynamic update mechanism that enables the teacher model to better guide the student model through adaptive updates.
- We propose a dynamic weight allocation method that allows the student model to focus on learning unfamiliar knowledge and reduce redundant information.
- We validate our approach on NLP tasks, outperforming baselines and providing implications for future model compression and optimization.

2 Related Work

2.1 Knowledge Distillation

In recent years, with the growing demand for large neural network compression [8], KD has received widespread attention as a lightweight technique. Knowledge distillation methods can be categorized into three types, namely response-based, feature-based, and relationship-based KD. Response-based KD was first proposed by Hinton [9], which uses the final output to transfer the knowledge of the tag. Previous work applied this idea from [10,3] to BERT, resulting in a smaller model with slightly reduced performance. For feature-based methods, knowledge is transferred by refining the internal representations of the teacher and student models. For example, TinyBERT

[11] extracts hidden states and attention distributions, while MINILM [12] distills self-attention distributions and value relationships, focusing on intermediate-layer representations to enhance the student model's learning ability. Relation-based KD methods transfer relationships between different layers or components of the teacher model. For example, CKD [6] distills structural knowledge by modeling token relationships in horizontal and vertical directions. AD-KD [7] explores distillation from an attribution perspective, analyzing the teacher's reasoning to transfer data-specific knowledge.

2.2 Dynamic knowledge distillation

In the field of NLP, Pro-KD [13] utilizes an adaptive temperature coefficient to enhance the distillation process by adjusting the output smoothness during training, thereby providing a crucial foundation for dynamic distillation. MetaDistil [14] leverages meta-learning techniques to update the teacher model, enabling it to better guide the student model. However, the weight allocation mechanisms of these methods still lack precise awareness of the student model's knowledge proficiency, which may result in the generation of redundant information. In the field of computer vision (CV), researchers have explored dynamic knowledge source distillation to enhance the diversity of knowledge. Adaptive-KD [15] achieves dynamic fusion of multi-layer features by adaptively adjusting the distillation weights of each layer; ATMKD [16] uses dynamic learnable temperature to adaptively control the difficulty of multi-teacher knowledge. These methods strengthen knowledge representation, but the teacher model cannot be adjusted based on student feedback, limiting its application potential in complex environments.

3 Methodology

Given a teacher model *T* and a student model *S* with parameters θ_T and θ_S, the input is sample *x* and the output truth value is *y*. Our goal is to design an efficient distillation framework that enables *S* to learn from *T* with limited computational resources. The overall framework is shown in **Fig. 1**.

3.1 Adaptive Distillation

Parameter Updates *S′*. In theory, we first use traditional knowledge distillation methods to update the parameters of the student model and evaluate its performance on an independent validation set. In practice, to prevent the student model from overfitting, we first copy the student model *S* as a student copy *S′* and update *S′*. Subsequently, gradient descent is used to pass feedback to the teacher model so that it can dynamically adjust to the state of the student model. The parameters of the student model are updated as follows:

$$\theta'_{S'}(\theta_T) = \theta_{S'} - \eta \nabla_{\theta_{S'}} \mathcal{L}_s(x; \theta_{S'}; \theta_T) \tag{1}$$

where η is the learning rate of the student model, and $\theta'_{s'}$ represents its updated parameters, the loss function $\mathcal{L}_S(x; \theta_{s'}; \theta_T)$ is defined as:

$$\mathcal{L}_S(x; \theta_{S'}; \theta_T) = \alpha \mathcal{L}_h(y, \theta_{S'}) + (1 - \alpha) \mathcal{L}_{KD}(\theta_{S'}, \theta_T) \tag{2}$$

where α is a hyperparameter that balances the relative importance of hard labeling loss $\mathcal{L}_h$ and knowledge distillation loss $\mathcal{L}_{KD}$.

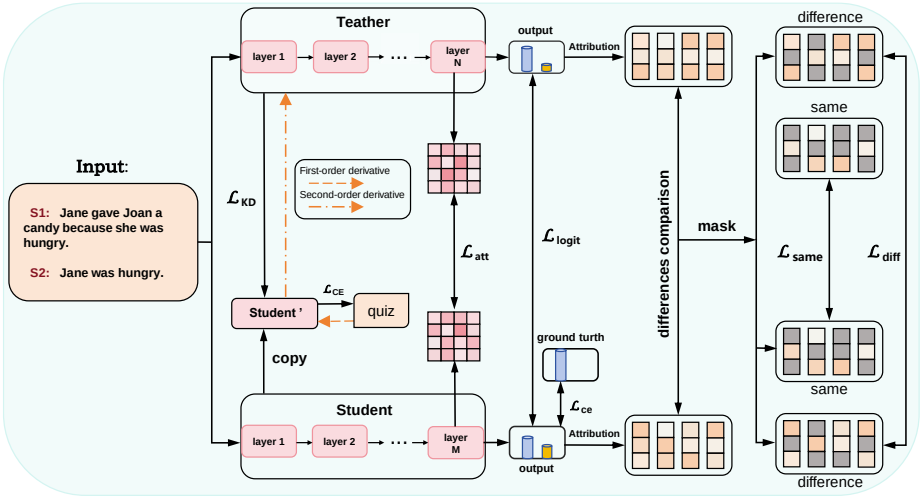

Fig. 1. Overview of our method framework.

Parameter Updates *T*. The updated student copy computes the loss function of the student model on the test dataset Q, which consists of a subset of questions taken from dev:

$$\mathcal{L}_q = \mathcal{L}_h\left(y, \theta_S'(\theta_T)\right) \tag{3}$$

We use this loss as a feedback signal to update teachers by computing second derivatives and gradient descent:

$$\theta_T' = \theta_T - \mu\nabla_{\theta_T}\mathcal{L}_h\left(y_q, \theta_S'(\theta_T)\right) \tag{4}$$

where μ is the learning rate of the teacher model.

Finally, we discard the experimental subject *S* and distill the real student model with the updated teacher model, through which the teacher model can dynamically adjust the knowledge expression and improve the knowledge transfer efficiency.

3.2 Dynamic Weight Allocation Policy

To optimize the learning effect of the student model on the saliency features of the teacher model, we used a dynamic weight adjustment policy. This strategy dynamically adjusts the weights according to the MSE of saliency differences to balance the attention of the student model to the same parts and different parts of the teacher model.

Attribution Calculator. The significance is calculated using the Integrated Gradients (IG) method [17], a gradient-based estimation method that quantifies the impact of

input features on model predictions by synthesizing the differences between input features and baseline features. Specifically, for the teacher model, the input gradient t_{input_grad} is obtained by calculating the gradient $t_{hidden}[0]$ of the hidden layer $t_{hidden}[0]$ with respect to the loss function t_{loss}. Then, the element-wise product of input_grad and $t_{hidden}[0]$ is taken, and the absolute value is calculated and normalized to obtain the significance S_T of the teacher model:

$$S_t = \frac{\left\| t_{\text{input\_grad}} \odot t_{\text{hidden}}[0] \right\|_2}{\left\| t_{\text{saliency}} \right\|_p} \tag{5}$$

where $\odot$ denotes the element-wise product, $\|\cdot\|_2$ denotes the *L2* norm, and p denotes the norm for normalization. For the student model, we can compute the student significance S_S using a similar method.

Weight Allocation. It is unreasonable to assign the same learning weight to the knowledge of the familiar and unfamiliar parts of the student model. To address this, we introduce a dynamic weighting strategy based on the knowledge familiarity of the student model. First, we use the saliency scores of the teacher model and the student model to calculate the difference *D*:

$$D = |S_t - S_s|. \tag{6}$$

To control the impact of *D*, we introduce a dynamic threshold θ with an initial value of θ_0, which decays over time with a rate γ:

$$\theta^{(t+1)} = \gamma\theta^{(t)}, \tag{7}$$

where *t* denotes the current training step. Based on this threshold, we construct two binary masks to distinguish between familiar and unfamiliar knowledge:

$$\text{same\_mask} = (D < \theta), \ \ \text{diff\_mask} = (D \geq \theta). \tag{8}$$

To dynamically balance the learning of familiar knowledge and unfamiliar knowledge, we adaptively update the weights w_{same} and w_{diff} according to the ratio of the number of activated elements N_s, and N_d in same_mask and diff_mask:

$$w_{diff} = max\left(w_{min}, \frac{N_d}{N_s+N_d}\right), \ \ w_{same} = min\left(w_{max}, \frac{N_s}{N_s+N_d}\right). \tag{9}$$

where w_{min} and w_{max} denote the minimum and maximum values of the weights, respectively. These parameters are determined by experimental validation to ensure the effectiveness of the weight adjustment strategy. Then the loss function of the similar part and the loss function of the dissimilar part are as follows:

$$\mathcal{L}_{same} = \frac{1}{N}\sum_{i=1}^{N}[(S_t \times same\_mask) - (S_s \times same\_mask)] \tag{10}$$

$$\mathcal{L}_{diff} = \frac{1}{N}\sum_{i=1}^{N}[(S_t \times diff\_mask) - (S_s \times diff\_mask)] \tag{11}$$

Finally, the knowledge distillation loss function $\mathcal{L}_{KD}$ can be expressed as:

$$\mathcal{L}_{kd} = (1-\alpha)\mathcal{L}_{ce} + \alpha\mathcal{L}_{logit} + \beta\left(w_{same} \times \mathcal{L}_{same} + w_{diff} \times \mathcal{L}_{diff}\right) \quad (12)$$

where $\mathcal{L}_{ce}$ is the cross-entropy loss concerning the truth value y and $\mathcal{L}_{logit}$ is the loss on the output logits of the teacher and student models. Where α and β are two hyperparameters.

Our method can more flexibly adapt to the learning progress of the student model, allowing the student model to focus on learning unfamiliar knowledge. In this way, we can effectively adjust the weights during the knowledge distillation process, thereby improving the performance of the student model.

3.3 Attention Distillation

The last layer of the attention matrix has rich high-level semantic patterns and long-distance dependencies, which are particularly effective in capturing the discriminative characteristics of the input data. So our method also utilizes the final layer attention matrices of the teacher and student models denoted as A^T and A^S respectively. By distilling this layer of the attention matrix, we enrich the cognitive dimension in the underdeveloped student model. So the loss function for attention matrix distillation is as follows:

$$\mathcal{L}_{att} = \mathcal{L}_{KL}(A_S, A_T). \quad (13)$$

3.4 Overall

In the above distillation framework, multiple modules collaborate to optimize the student model. To unify their goals, we define a total loss function $\mathcal{L}_{total}$ to improve performance under limited computing resources:

$$\mathcal{L}_{total} = \lambda_S\mathcal{L}_S + \lambda_{att}\mathcal{L}_{att} + \lambda_{kd}\mathcal{L}_{kd}. \quad (14)$$

The parameters λ_S, λ_{att}, λ_{kd} are hyperparameters used to balance the relative importance of loss functions of different modules. By minimizing $\mathcal{L}_{total}$, our goal is to enable the student model to better learn the performance of the teacher model under limited computational resources.

4 Experiments

4.1 Datasets And Baselines

Datasets. We evaluate our method on eight tasks on the GLUE benchmark [18], covering a range of different language understanding challenges. These include two single-sentence tasks, three similarity and paraphrase tasks, and three inference tasks. Following standard practice, we use the official evaluation metrics for each task, including accuracy, F1 score, Matthews correlation coefficient, and Spearman rank correlation, which is consistent with previous studies.

Baselines. We compare our method with several state-of-the-art baseline methods, including response-based, feature-based, and relation-based KD methods. Response-based methods include Vanilla KD [9] and PD [3]. Feature and relation-based methods include PKD [4] for intermediate representations, TinyBERT [11] for attention matrices, and CKD [6] and MGSKD [5] for relational knowledge. In addition, we also include ADKD [7], which incorporates attribution knowledge into the distillation process to improve the efficiency of knowledge distillation.

4.2 Implementation Details

Our experiments were implemented by Pytorch and the Huggingface Transformers library [19]. We fine-tuned $BERT_{base}$ as the teacher model θ_T and used asmaller BERT variant with 6 transformer layers, 768 hidden neurons, and 12 attention heads as the student model θ_s. For hyperparameter tuning, we searched for the optimal student model learning rate η, in $\{2e-5, 3e-5, 4e-5\}$, the optimal teacher model learning rate μ in $\{1e-5, 2e-5, 3e-5\}$, the optimal α in $\{0.5, 0.6, 0.7, 0.8\}$, and the optimal β in $\{0.5, 1.0, 1.5, 2.0\}$. The initial value of the dynamic threshold θ_0 is set to 0.2 and the decay rate γ is 0.9. For w_{min} and w_{max}, we set w_{min} = 0.5 and w_{max} = 0.5 according to the ablation experiment verification in Section 5.7 to ensure the efficiency of the weight allocation policy. The weighting coefficients λ_S, λ_{att}, and λ_{kd} are tuned in the range of $\{0.2, 0.5, 0.8, 1.0\}$.

5 Results and Analysis

5.1 Main Results

We report the experimental results on the development and test sets of eight GLUE tasks [20] in **Table 1**. Overall, our method outperforms all baselines on most datasets. Specifically, on the development set, our method improves on average by 0.6 points over the second-best method ADKD [7]; on the test set, it also improves on ADKD by 0.6 points. In addition, the experimental results show that our method does not perform well on the SST-2 dataset. We believe that this is mainly because the sentences in the SST-2 task are shorter, and the soft labels output by the teacher model already contain most of the knowledge, so the room for improvement of our method is relatively limited.

5.2 Ablation Study

From the ablation experiment results in **Table 2**, it can be seen that the model performance is significantly reduced after removing the adaptive update module, indicating that the dynamic adjustment of the teacher model plays a key role in improving the knowledge transfer effect. When the dynamic weight allocation policy is canceled, the performance also drops significantly, indicating that this strategy is of great value in guiding the student model to reasonably allocate attention to familiar and unfamiliar knowledge. In contrast, removing the attention distillation module only brings a slight performance degradation. Although there is an impact, it is not as significant as the previous two.

In summary, the adaptive update mechanism and dynamic weight allocation policy of the teacher model play a vital role in improving the knowledge distillation effect.

Table 1. Overall results on the GLUE benchmark, except for Vanilla KD and AD-KD, are from [7]. Test scores are from the official GLUE server. Averages exclude MNLI-mm; best and second-best student results are bolded and underlined.

Method	Params	Evaluation Metrics								Avg
		CoLA (Mcc)	MNLI-m/mm (Acc)	SST-2 (Acc)	QNLI (Acc)	MRPC (F1)	QQP (Acc)	RTE (Acc)	STS-B (Spear)	
Dev										
(T)BERT$_{base}$	110M	60.1	84.7/84.4	93.5	91.5	91.3	91.5	69.7	89.4	84.0
(S)BERT$_6$	66M	51.2	81.3/82.3	90.7	89.3	89.2	90.4	65.9	88.2	80.8
Vanilla KD[9]	66M	53.4	82.6/82.9	91.2	90.3	89.0	90.5	67.4	88.6	81.6
PD[3]	66M	-	82.5/83.4	91.1	89.4	89.4	90.7	66.7	-	-
PKD[4]	66M	45.5	81.3/-	91.3	88.4	85.7	88.4	66.5	86.2	79.2
TinyBERT [11]	66M	53.8	83.1/83.4	92.3	88.9	88.8	90.5	66.9	88.3	81.7
CKD [6]	66M	55.1	83.6/84.1	**93.0**	90.5	89.6	91.2	67.3	**89.0**	82.4
MSKGD [5]	66M	49.1	83.3/83.9	91.7	90.3	89.8	91.2	67.9	88.5	81.5
AD-KD[7]	66M	**57.5**	82.5/83.2	91.7	91.0	**90.8**	91.0	69.2	88.3	82.6
Ours	66M	57.3	**83.8**/**84.3**	91.5	**91.5**	90.5	**91.4**	**70.7**	**89.0**	**83.2**
Test										
(T)BERT$_{base}$	110M	51.4	84.3/84.0	93.9	90.8	87.6	89.2	67.6	85.3	81.2
(S)BERT$_6$	66M	41.6	81.7/80.8	91.3	88.7	85.2	88.0	63.9	82.1	77.8
Vanilla KD[9]	66M	42.2	82.6/81.8	92.0	89.1	86.3	88.2	65.1	82.5	78.5
PD[3]	66M	-	82.8/82.2	91.8	88.9	86.8	88.9	65.3	-	-
PKD[4]	66M	43.5	81.5/81.0	92.0	89.0	85.0	88.9	65.5	81.6	78.4
MSKGD [5]	66M	42.8	83.4/**82.8**	**92.1**	89.5	**87.0**	**89.1**	63.7	82.2	78.7
AD-KD[7]	66M	**45.8**	82.3/81.9	91.6	89.8	86.8	88.8	65.6	82.6	79.1
Ours	66M	45.6	**83.8**/82.6	91.3	**90.6**	**87.0**	89.0	**66.7**	**83.8**	**79.7**

5.3 Student Model Size

To evaluate the effectiveness of different student sizes, we compare our method with vanilla KD [3] on MRPC and QNLI. BERT-Base is compressed into BERT-6L, Medium, Small, Mini, and Tiny. The performance versus the number of student parameters is shown in **Fig. 2**. Our method consistently outperforms vanilla KD, demonstrating its effectiveness and robustness.

5.4 Attention Distillation Layer

To determine the optimal attentional distillation layer, we systematically compared the effects of distillation from different layers. As shown in **Fig. 3**, using the twelfth layer

of the BERT attention matrix yields significantly better performance than using middle or lower layers on both the SST-2 sentiment classification task and the STS-B semantic similarity task. This improvement is attributed to the stronger semantic abstraction capability of the final layer, which tends to focus on task-relevant keywords. In contrast, lower layers attend more to structural or less informative tokens, resulting in reduced accuracy. These findings suggest that the final attention layer encodes richer high-level semantic information, which is more beneficial for downstream decision-making.

5.5 Impact of Teacher Model Updates

To validate the instructional benefits of teacher model updating, we compared static and dynamic teacher attribution maps and their effects on student learning. As shown in **Fig. 4**, the information conveyed by the static teacher model may be confusing to the student model, resulting in erroneous judgments from the student model. However, dynamic teachers significantly increase their attention to the points that are confounded by students through an adaptive updating mechanism. The ablation experiment further demonstrates that student accuracy decreases by 2.6 percentage points when no adaptive updating of the teacher model is performed. This suggests that dynamic teachers achieve more efficient knowledge transfer through adaptive updating by both improving knowledge representation and pinpointing student needs.

Table 2. Ablation study based on GLUE development set.

Method	CoLA (Mcc)	MNLI-m/mm (Acc)	SST-2 (Acc)	QNLI (Acc)	MRPC (F1)	QQP (Acc)	RTE (Acc)	STS-B (Spear)
Ours	57.3	83.8/84.3	91.5	91.5	90.5	91.4	70.7	89.0
w/o Update	56.8	82.7/83.3	91.0	91.1	90.1	91.1	69.8	88.7
w/o Attention	57.2	83.6/84.1	91.3	91.4	90.3	91.2	70.4	88.9
w/o Dynamic Weight	56.9	83.2/83.7	91.2	91.2	89.9	90.9	69.5	88.5

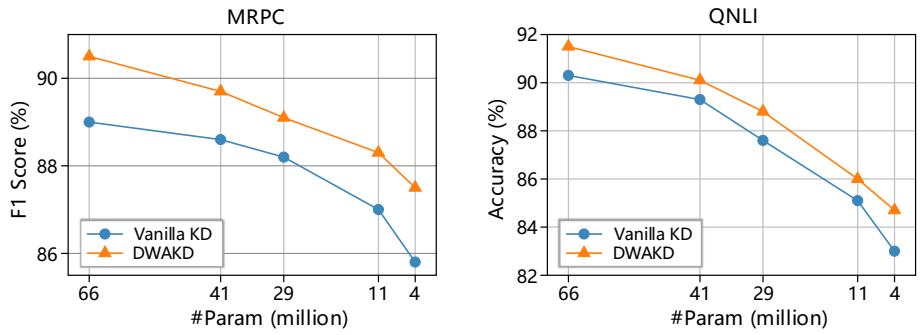

Fig. 2. Results for different student sizes.

5.6 Impact of θ_0 and γ

To distinguish between acquired and unacquired knowledge, we introduce a threshold θ and an attenuation rate γ. **Fig. 5** shows the results of varying θ_0 and γ on MRPC and QNLI, with one fixed while the other is adjusted. When θ_0 is too large, the student model tends to overlook signifcant differences, with accuracy/Fl scores peaking at $\theta_0 = 0.2$ and then gradually decreasing. For γ, performance follows a rising and then falling trend as γ increases. A low γ causes the threshold to decay too quickly, reducing targeted guidance and leading the model to a local optimum. On the other hand, a high γ results in slow decay, retaining a high threshold that inhibits convergence speed. The best performanceis achieved at $\gamma = 0.9$.

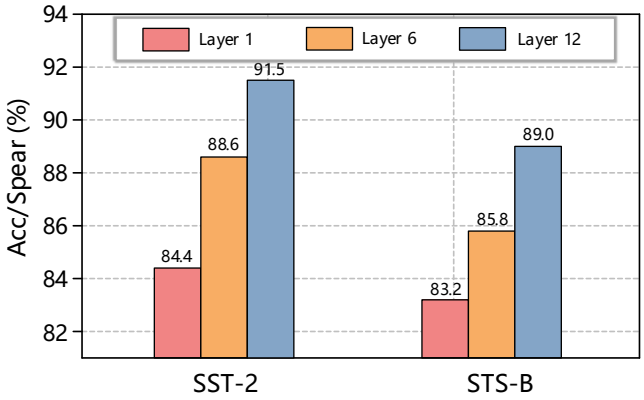

Fig. 3. Results of using different attention layers.

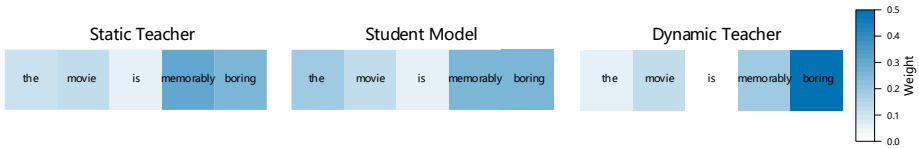

Fig. 4. Attribution weights for static and dynamic teacher models based on student model states.

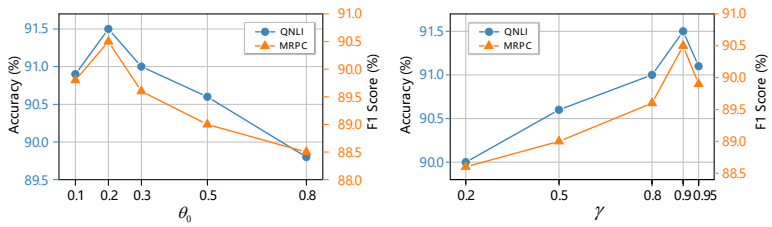

Fig. 5. Results of using different θ_0 and γ.

5.7 Impact of w_{min} and w_{max}

To explore the impact of the optimal configuration of w_{min} and w_{max} on the performance of the student model, we designed an ablation experiment to compare the model performance under different weight settings. The experimental results are shown in **Table 3**. When $w_{min} = w_{max} = 0.5$ the student model performs best on both tasks. If w_{min} is set too small, the model may ignore the difference area, resulting in insufficient learning of unfamiliar knowledge, thereby reducing the accuracy; when the w_{diff} value is too large, the student model may try to overfit those areas that it is not familiar with, This over-adaptation may lead to overfitting, thereby sacrificing the generalization ability of the model, and in actual training, setting an appropriate lower limit can ensure that the student model always focuses on the difference part, while limiting w_{same} can prevent the model from over-focusing on the familiar part and improve its generalization ability, The results verify that our proposed "focus on learning unfamiliar knowledge" policy shows stability and versatility in different tasks.

Table 3. Results of different w_{min} and w_{max} settings.

w_{min}/w_{max}	MNLI Dev (Acc)	MNLI Test (Acc)	QQP Dev (Acc)	QQP Test (Acc)
0.3 / 0.7	82.5	82.2	89.8	88.0
0.4 / 0.6	83.0	82.8	90.7	88.4
0.5 / 0.5	**83.8**	**83.8**	**91.4**	**89**
0.6 / 0.4	82.0	81.7	90.5	88.1

6 Conclusion

In this paper, we propose a novel dynamic knowledge distillation framework to improve the efficiency and effectiveness of pre-trained language model compression. Unlike traditional methods where the student model passively imitates the teacher, our framework supports two-way interaction: the teacher model dynamically adjusts its output based on the student's reverse gradient feedback. We introduce a dynamic weight allocation policy that adjusts the learning focus according to the student's mastery of different knowledge components, thereby emphasizing less familiar areas for more targeted distillation. In addition, to help the student model understand richer semantic information, we also distill the last layer of the attention matrix. Our method combines an adaptive teacher update mechanism with a dynamic weight allocation policy to more effectively guide the learning of the student model. Extensive experiments, including ablation studies, demonstrate the superior performance of our method and verify the contribution of each component.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Grant 62373132 , in part by the S&T Program of Hebei under Grant 236Z1602G, in part by the S&T Program of Shijiazhuang under Grant 241791367A, in part by the S&T

Program of Baoding under Grant 2472P006, in part by the Excellent Youth Research Innovation Team of Hebei University under Grant QNTD202411, in part by the Interdisciplinary Research Program of Hebei University under Grant DXK202409, in part by the Central Government for Local Science and Technology Development under Grant 246Z0704G, in part by the Innovation Capacity Enhancement Program-Science and Technology Platform Project of Hebei Province under Grant 22567638H, in part by High-level Talent Scientific Research Start Foundation of Hebei University under Grant 521100223209.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), pages 4171–4186. Association for Computational Linguistics, 2019.
2. Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. CoRR, abs/1907.11692, 2019.
3. Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Well-read students learn better: The impact of student initialization on knowledge distillation. CoRR, abs/1908.08962, 2019.
4. Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for BERT model compression. CoRR, abs/1908.09355, 2019.
5. Chang Liu, Chongyang Tao, Jiazhan Feng, and Dongyan Zhao. Multi-granularity structural knowledge distillation for language model compression. InAnnual Meeting of the Association for Computational Linguistics, 2022.
6. Geondo Park, Gyeongman Kim, and Eunho Yang. Distilling linguistic context for language model compression. CoRR, abs/2109.08359, 2021.
7. Siyue Wu, Hongzhan Chen, Xiaojun Quan, Qifan Wang, and Rui Wang. AD-KD: Attribution-driven knowledge distillation for language model compression. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 8449–8465, July 2023.
8. Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. Bertof-theseus: Compressing BERT by progressive module replacing. CoRR, abs/2002.02925, 2020.
9. Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.
10. Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR, abs/1910.01108, 2019.
11. Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. CoRR, abs/1909.10351, 2019.

12. Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. CoRR, abs/2002.10957, 2020.
13. Mehdi Rezagholizadeh, Aref Jafari, Puneeth Salad, Pranav Sharma, Ali Saheb Pasand, and Ali Ghodsi. Pro-kd: Progressive distillation by following the footsteps of the teacher. CoRR, abs/2110.08532, 2021.
14. Wangchunshu Zhou, Canwen Xu, and Julian J. McAuley. Meta learning for knowledge distillation. CoRR, abs/2106.04570, 2021.
15. Sumanth Chennupati, Mohammad Mahdi Kamani, Zhongwei Cheng, and Lin Chen. Adaptive distillation: Aggregating knowledge from multiple paths for efficient distillation. CoRR, abs/2110.09674, 2021.
16. Yu-e Lin, Shuting Yin, Yifeng Ding, and Xingzhu Liang. Atmkd: adaptive temperature guided multi-teacher knowledge distillation. Multimedia Syst., 30(5), 2024.
17. Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. CoRR, abs/1703.01365, 2017.
18. Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. CoRR, abs/1804.07461, 2018.
19. Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. CoRR, abs/1910.03771, 2019.
20. Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. CoRR, abs/1804.07461, 2018.