

# Dynamic Knowledge Distillation for Pre-trained Language Models

Lei Li<sup>†</sup>, Yankai Lin<sup>§</sup>, Shuhuai Ren<sup>†</sup>, Peng Li<sup>§</sup>, Jie Zhou<sup>§</sup>, Xu Sun<sup>†</sup>

<sup>†</sup>MOE Key Laboratory of Computational Linguistics, School of EECS, Peking University

<sup>§</sup>Pattern Recognition Center, WeChat AI, Tencent Inc., China

{lilei, shuhuai\_ren}@stu.pku.edu.cn xusun@pku.edu.cn

{yankailin, patrickpli, withtomzhou}@tecent.com

## Abstract

Knowledge distillation (KD) has been proved effective for compressing large-scale pre-trained language models. However, existing methods conduct KD statically, e.g., the student model aligns its output distribution to that of a selected teacher model on the pre-defined training dataset. In this paper, we explore whether a dynamic knowledge distillation that empowers the student to adjust the learning procedure according to its competency, regarding the student performance and learning efficiency. We explore the dynamical adjustments on three aspects: teacher model adoption, data selection, and KD objective adaptation. Experimental results show that (1) proper selection of teacher model can boost the performance of student model; (2) conducting KD with 10% informative instances achieves comparable performance while greatly accelerates the training; (3) the student performance can be boosted by adjusting the supervision contribution of different alignment objective. We find dynamic knowledge distillation is promising and provide discussions on potential future directions towards more efficient KD methods.<sup>1</sup>

## 1 Introduction

Knowledge distillation (KD) (Hinton et al., 2015) aims to transfer the knowledge from a large teacher model to a small student model. It has been widely used (Sanh et al., 2019; Jiao et al., 2020; Sun et al., 2019) to compress large-scale pre-trained language models (PLMs) like BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) in recent years. By knowledge distillation, we can obtain a much smaller model with comparable performance, while greatly reduce the memory usage and accelerate the model inference.

Although simple and effective, existing methods usually conduct the KD learning procedure

statically, e.g., the student model aligns its output probability distribution to that of a selected teacher model on the entire pre-defined corpus. In other words, the following three aspects of KD are specified in advance and remain unchanged during the learning procedure: (1) the teacher model to learn from (learning target); (2) the data used to query the teacher (learning material); (3) the objective functions and the corresponding weights (learning method). However, as the student competency evolves during the training stage, it is particularly unreasonable to pre-define these learning settings and keep them unchanged. Conducting KD statically may lead to (1) unqualified learning from a too large teacher, (2) repetitive learning on instances that the student has mastered, and (3) sub-optimal learning on alignments that are unnecessary. This motivates us to explore an interesting problem: whether a dynamic KD framework considering the student competency evolution during training can bring benefits, regarding the student performance and learning efficiency?

In this paper, we propose a dynamic knowledge distillation (Dynamic KD) framework, which attempts to empower the student to adjust the learning procedure according to its competency. Specifically, inspired by the success of active learning (Settles, 2009), we take the prediction uncertainty, e.g., the entropy of the predicted classification probability distribution, as a proxy of the student competency. We strive to answer the following research questions: (RQ1) Which teacher is proper to learn as the student evolves? (RQ2) Which data are actually useful for student models in the whole KD stage? (RQ3) Does the optimal learning objective change in the KD process? In particular, we first explore the impact of the teacher size to dynamic knowledge distillation. Second, we explore whether dynamically choosing instances that the student is uncertain for KD can lead to a better performance and training efficiency trade-off. Third,

<sup>1</sup>Our code is available at <https://github.com/lancopku/DynamicKD>

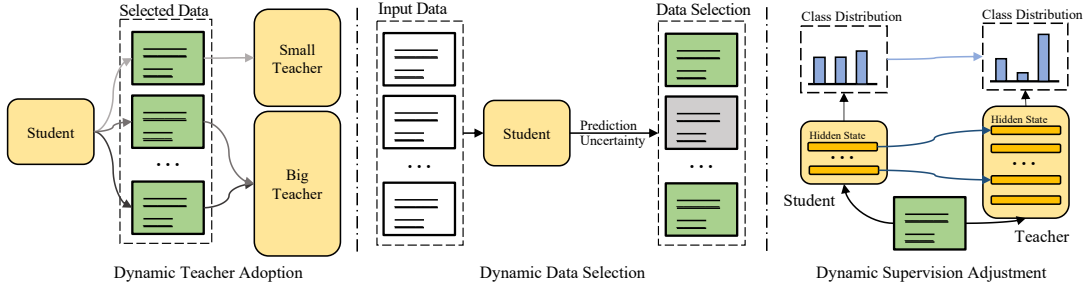


Figure 1: The three aspects of dynamic knowledge distillation explored in this paper. Best viewed in color.

we explore whether the dynamic adjustment of the supervision from alignment of prediction probability distributions and the alignment of intermediate representations in the whole KD stage can improve the performance.

Our experimental results show that: (1) A larger teacher model with more layers may raise a worse student. We show that selecting the proper teacher model according to the competency of the student can improve the performance. (2) We can achieve comparable performance using only 10% informative instances selected according to the student prediction uncertainty. These instances also evolve during the training as the student becomes stronger. (3) We can boost the student performance by dynamically adjusting the supervision from different alignment objectives of the teacher model.

Our observations demonstrate the limitations of the current static KD framework. The proposed uncertainty-based dynamic KD framework only makes the very first attempt, and we are hoping this paper can motivate more future research towards more efficient and adaptive KD methods.

## 2 Background: Knowledge Distillation

Given a student model  $S$  and a teacher model  $T$ , knowledge distillation aims to train the student model by aligning the outputs of the student model to that of the teacher. For example, Hinton et al. (2015) utilize the teacher model outputs as soft targets for the student to learn. We denote  $S(x)$  and  $T(x)$  as the output logit vector of the student and the teacher for input  $x$ , respectively. The KD can be conducted by minimizing the Kullback-Leibler (KL) divergence distance between the student and teacher prediction:

$$\mathcal{L}_{KL} = \text{KL}(\sigma(S(x)/\tau) \parallel \sigma(T(x)/\tau)), \quad (1)$$

where  $\sigma(\cdot)$  denotes the softmax function and  $\tau$  is a temperature hyper-parameter. The student param-

eters are updated according to the KD loss and the original classification loss, i.e., the cross-entropy over the ground-truth label  $y$ :

$$\mathcal{L}_{CE} = -y \log \sigma(S(x)), \quad (2)$$

$$\mathcal{L} = (1 - \lambda_{KL})\mathcal{L}_{CE} + \lambda_{KL}\mathcal{L}_{KL}, \quad (3)$$

where  $\lambda_{KL}$  is the hyper-parameter controlling the weight of knowledge distillation objective. Recent explorations also find that introducing KD objectives of alignments between the intermediate representations (Romero et al., 2015; Sun et al., 2019) and attention map (Jiao et al., 2020; Wang et al., 2020) is helpful. Note that conventional KD framework is static, i.e., the teacher model is selected before KD and the training is conducted on all training instances indiscriminately according to the predefined objective and the corresponding weights of different objectives. However, it is unreasonable to conduct the KD learning procedure statically as the student model evolves during the training. We are curious whether adaptive adjusting the settings on teacher adoption, dataset selection and supervision adjustment can bring benefits regarding student performance and learning efficiency, motivating us to explore a dynamic KD framework.

## 3 Dynamic Knowledge Distillation

In this section, we introduce the dynamic knowledge distillation. The core idea behind is to empower the student to adjust the learning procedure according to its current state, and we investigate the three aspects illustrated in Figure 1.

### 3.1 Dynamic Teacher Adoption

The teacher model plays a vital role in KD, as it provides the student soft-targets for helping the student learn the relation between different classes (Hinton et al., 2015). However, there are few investigations regarding how to select a proper teacher for the

Method	RTE	IMDB	CoLA	Avg.
BERT <sub>BASE</sub>	67.8	89.1	54.2	70.4
BERT <sub>LARGE</sub>	72.6	90.4	60.1	74.4
No KD	63.7	86.3	39.0	63.0
KD w/ BERT <sub>BASE</sub>	64.9	86.9	39.4	63.7
KD w/ BERT <sub>LARGE</sub>	64.5	86.5	38.2	63.1
KD w/ Ensemble	64.9	86.7	39.9	63.8
Uncertainty-Hard	<b>66.9*</b>	86.3	<b>42.7*</b>	<b>65.3</b>
Uncertainty-Soft	66.4*	<b>87.1*</b>	41.0	64.8

Table 1: We find that bigger teacher with better performance raises a worse student model. Results are average of 3 seeds on the validation set. \* denotes statistically significant improvement over the best performing baseline with  $p < 0.05$ .

student in KD for PLMs during the training dynamically. In the KD of PLMs, it is usually all teacher models are with the same model architecture, i.e., huge Transformer (Vaswani et al., 2017) model. Thus, the most informative factor of teacher models is their model size, e.g., the layer number of the teacher model and the corresponding hidden size. This motivates us to take a first step to explore the impact of model size.

### 3.1.1 Bigger Teacher Not Always Raises Better Student

Specifically, we are curious about whether learning from a bigger PLM with better performance can lead to a better distilled student model. We conduct probing experiments to distill a 6-layer student BERT model from BERT<sub>BASE</sub> with 12 layers, and BERT<sub>LARGE</sub> with 24 layers, respectively. We conduct the experiment on two datasets, RTE (Bentivogli et al., 2009) and CoLA (Warstadt et al., 2019), where two teacher models exhibit clear performance gap, and a sentiment classification benchmark IMDB (Maas et al., 2011). Detailed experimental setup can be found in Appendix A.

As shown in Table 1, we surprisingly find that while the BERT<sub>LARGE</sub> teacher clearly outperforms the small BERT<sub>BASE</sub> teacher model, the student model distilled by the BERT<sub>BASE</sub> teacher achieves better performance on all three datasets. This phenomenon is counter-intuitive as a larger teacher is supposed to provide better supervision signal for the student model. We think that there are two possible factors regarding the size of teacher model that leading to the deteriorated performance:

(1) The predicted logits of the teacher model become less soft as the teacher model becomes larger and more confident about its prediction (Guo et al.,

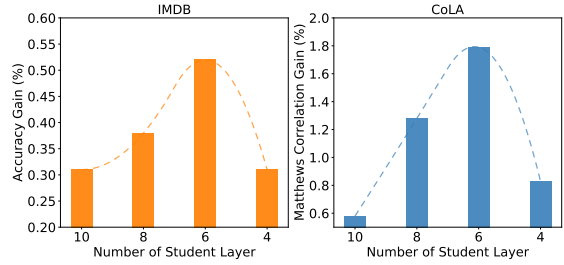


Figure 2: Performance gain of distilled student model with various layer sizes. The teacher model is BERT<sub>BASE</sub> with 12 layers.

2017; Desai and Durrett, 2020), which decreases the effect of knowledge transfer via the soft targets. We find that a smaller  $\tau$  also leads to a decreased performance of the student model, indicating the the less-softened teacher prediction will decrease the student performance.<sup>2</sup>

(2) The capacity gap between the teacher and student model increases as the teacher becomes larger. The competency of the student model can not match that of the large teacher model, which weakens the performance of KD.

To explore the combined influence of these factors, we distill student models with different layers and plot the performance gain compared to directly training the student model without distillation in Figure 2. It can be found that by decreasing the student size, the better supervision from teacher model boosts the performance, while the two counteractive factors dominate as the gap becomes much larger, decreasing the performance gain. We notice that this phenomenon is also observed by Mirzadeh et al. (2020) in computer vision tasks using convolutional networks, showing that it is a widespread issue and needs more in-depth investigations. Note that BERT<sub>BASE</sub> and BERT<sub>LARGE</sub> also differs from the number of hidden size, the experiments regarding the hidden size, where the phenomenon also exists and corresponding results can be found in Appendix C.

### 3.1.2 Uncertainty-based Teacher Adoption

Our preliminary observations demonstrate that selecting a proper teacher model for KD is significant for the student performance. While the capacity gap is an inherent problem once the teacher and the student are set, we are curious about whether dynamically querying the proper teacher according to the student competency during training can make

<sup>2</sup>Refer to Appendix B for details.

the full use of teacher models. Without loss of generality, we conduct KD to train a student model from two teacher models with different numbers of Transformer layers. We assume that during the initial training stage, the student can rely more on the small teacher model, while turns to the large teacher for more accurate supervision when it becomes stronger. Specifically, we propose to utilize the student prediction uncertainty as a proxy of the competency, inspired by the successful applications in active learning (Settles, 2009), and design two uncertain-based teacher adoption strategies:

**Hard Selection:** The instances in one batch are sorted according to the student prediction uncertainty, i.e., the entropy of predicted class distribution. Then the instances are evenly divided into instances that the student most uncertain about and instances that model is most confident about. For the uncertain part, the small teacher is queried for supervision signals, while the large teacher provides the soft-label for the instances that the student is confident about.

**Soft Selection:** The corresponding KD loss weights from two teachers are adjusted softly at instance-level. Formally, given two teacher model  $T_1$  (BERT<sub>BASE</sub>, in our case) and  $T_2$  (BERT<sub>LARGE</sub>), we can re-write the KD objective in the multiple teacher setting as:

$$\mathcal{L}_{KD} = w_1 \mathcal{L}_{KL}^{T_1} + w_2 \mathcal{L}_{KL}^{T_2} \quad (4)$$

where  $\mathcal{L}_{KL}^T$  denotes matching loss of the output logits of the student model and the teacher model  $t$ . The  $w_1$  and  $w_2$  controls the relative contribution of the supervisions from the two teachers. We adaptively down-weight the supervision from the large teacher when the student are uncertain about the training instances. The prediction uncertainty is adopted as a measurement of the student competency for instance  $x$ :

$$u_x = \text{Entropy}(\sigma(S(x))) \quad (5)$$

where  $\sigma$  is a normalization function, e.g, softmax function for mapping the logit vector to probability distribution. The  $w_1$  and  $w_2$  are adjusted as follows:

$$w_1 = \frac{u_x}{U}, \quad w_2 = 1 - \frac{u_x}{U} \quad (6)$$

where  $U$  is a normalization factor which re-scales the weight to  $[0, 1]$ . In this way, the student will pay more attention to the small teacher when it is

uncertain about the current instance, while relies on the large teacher when it is confident about its prediction.

### 3.1.3 Experiments

**Settings** We conduct experiments to distill a 6-layer student model from BERT<sub>BASE</sub> and BERT<sub>LARGE</sub>, on RTE, CoLA and IMDB, following settings of probing analysis in Section 3.1.1.

**Results** The results of the proposed selection strategies are shown in Table 1. We observe that the hard selection strategy achieves an overall 65.3 accuracy which outperforms directly learning from the ensemble of two teacher models. This demonstrates that the proposed strategy is effective by selecting the proper teacher model to learn. The soft selection strategy also outperforms the baseline while falls little behind with the hard version. We attribute it to that the provided supervisions of two teachers are of different softness, thus may confuse the student model.

## 3.2 Dynamic Data Selection

The second research question we want to explore is which data will be more beneficial for the performance of the student. As the distillation proceeds, the student is becoming stronger, thus the repetitive learning on the instances those the student has mastered on can be eliminated. If there are such instances that are vital for the learning of the student model, can we only conduct KD on these instances, for improving the learning efficiency? Besides, do the vital instances remain static or also evolve with the student model? These questions motivate us to explore the effect of dynamically selecting instances.

### 3.2.1 Uncertainty-based Data Selection

We propose to actively select informative instances according to student prediction uncertainty, inspired by the successful practice of active learning (Settles, 2009). Formally, given  $N$  instances in one batch, for each instance  $x$ , the corresponding output class probability distribution over the class label  $y$  of the student model is  $P(y | x) = \sigma(S(x))$ . We compute an uncertainty score  $u_x$  for  $x$  using the follow strategies with negligible computational overhead:

**Entropy** (Settles, 2009), which measures the uncertainty of the student prediction distribution:

$$u_x = - \sum_y P(y | x) \log P(y | x). \quad (7)$$

Method	#FLOPs	SST-5	IMDB	MRPC	MNLI-m / mm
BERT <sub>BASE</sub> (Teacher)	-	52.0	89.1	86.8	84.0 / 84.4
Vanilla KD	45.1B	47.4	86.8	80.2	81.7 / 82.0
Random	22.6B	46.8	86.4	79.7	81.4 / 81.6
Uncertainty-Entropy	28.2B	46.7	86.8	79.4	81.5 / 82.0
Uncertainty-Margin	28.2B	46.6	86.8	79.4	81.4 / 81.9
Uncertainty-LC	28.2B	46.5	86.8	79.4	81.4 / 81.9
$\Delta$	-	-0.6	0.0	-0.5	-0.2 / 0.0

Table 2: Dynamic data selection results with  $r$  set to 0.5. Results are averaged of 3 seeds on the validation set.  $\Delta$  denotes the minimal performance degradation of different selection strategies compares to vanilla KD.

Dataset	# Train	# Aug Train	# Dev	# Test	# Class
SST-5	8.8k	176k	1.1k	2.2k	5
IMDB	20k	400k	5k	25k	2
MNLI	393k	786,0k	20k	20k	3
MRPC	3.7k	74k	0.4k	1.7k	2
RTE	2.5k	50k	0.3k	3k	2
CoLA	8.5k	170k	1k	1k	2

Table 3: Statistics of datasets. # Aug Train denotes the number of the augmented training dataset following Jiao et al. (2020).

**Margin**, which is computed as the margin between the first and second most probable class  $y_1^*$  and  $y_2^*$ :

$$u_x = P(y_1^* | x) - P(y_2^* | x). \quad (8)$$

**Least-Confidence (LC)**, which indicates how uncertain the model about the predicted class  $\hat{y} = \arg \max_y P(y | x)$ :

$$u_x = 1 - P(\hat{y} | x) \quad (9)$$

We rank the instances in a batch according to its prediction uncertainty, and only choose the top  $N \times r$  instances to query the teacher model, where  $r \in (0, 1]$  is the selection ratio controlling the number to query. Note that in binary classification tasks like IMDB, the selected subsets using the above strategies are identical. We also design a **Random** strategy that selects  $N \times r$  instances randomly, to serve as a baseline for evaluating the effectiveness of selection strategies.

### 3.2.2 Experiments

**Settings** We conduct the investigation experiments on two sentiment classification datasets IMDB (Maas et al., 2011) and SST-5 (Socher et al., 2013), and natural language inference tasks including MRPC (Dolan and Brockett, 2005) and MNLI (Williams et al., 2018). The statistics of dataset and the implementation details can be found

in Table 3 and D, respectively. We report accuracy as the performance measurement for all the evaluated tasks.

Besides, we also provide the corresponding computational FLOPs for comparing the learning efficiency. In more detail, we divide the computational cost  $C$  of KD into three parts: student forward  $F_s$ , teacher forward  $F_t$  and student backward  $B_s$  for updating parameters. Note that  $F_s \approx B_s \ll F_t$ , as the teacher model is usually much larger than the student model. By actively learning only from  $N \times r$  instances that the student are most uncertain about, the cost is reduced to:

$$C' = N \times F_s + N \times r \times B_s + N \times r \times F_t. \quad (10)$$

For example, the number of computational FLOPs of a 6-layer student BERT model is 11.3B while that of a 12-layer BERT teacher model is 22.5B (Jiao et al., 2020). When  $r$  is set to 0.1, the total KD cost is reduced from 45.1B to 14.7B.

**Results with Original Dataset** The results when  $r$  set to 0.5 are listed in Table 2. Overall, it can be found that selecting the instances only lead negligible degradation of the student performance, compared to that of Vanilla KD, showing the effectiveness of the uncertainty-based strategies. Interestingly, the random strategy perform closely to the uncertainty-based strategies, which we attribute to that the underlying informative data space can also be covered by random selected instances. Besides, we notice that performance drop is smaller on the tasks with larger training data size. For example, selecting informative instances with prediction entropy leads to 0.2 accuracy drop on the MNLI dataset consisting 393k training instances, while causes 0.8 performance drop on MRPC with 3.7k instances. A possible reason is that for the tiny dataset, the underlying data distribution is not well covered by the training data, therefore further down-sampling the training data results in a larger performance gap. To verify this, we turn to the the setting where the original training dataset is enriched with augmentation techniques.

**Results with Augmented Dataset** Following TinyBERT (Jiao et al., 2020), we augment the training dataset 20 times with BERT mask language prediction, as it has been prove effective for distilling a powerful student model. Our assumption is that with the data augmentation technique, the training set can sufficiently cover the possible data

Method	#FLOPs	SST-5	IMDB	MRPC	MNLI-m / mm	Avg. ( $\uparrow$ )	$\Delta$ ( $\downarrow$ )
BERT <sub>BASE</sub> (Teacher)	-	53.7	88.8	87.5	83.9 / 83.4	79.5	-
TinyBERT <sup>†</sup>	24.9B	-	-	86.4	82.5 / 81.8	-	-
TinyBERT	24.9B	51.4	87.6	86.2	82.6 / 82.0	78.0	0.0
Random	2.49B	51.1	87.0	83.3	80.8 / 80.5	76.5	1.5
Uncertainty-Entropy	4.65B	51.5	<b>87.7</b>	<b>86.5</b>	<b>81.8</b> / 81.0	<b>77.7</b>	<b>0.3</b>
Uncertainty-Margin	4.65B	<b>51.6</b>	<b>87.7</b>	<b>86.5</b>	81.6 / <b>81.1</b>	<b>77.7</b>	<b>0.3</b>
Uncertainty-LC	4.65B	51.2	<b>87.7</b>	<b>86.5</b>	81.4 / 80.8	77.5	0.5

Table 4: Test results when the selection ratio  $r = 0.1$  for dynamic data selection on various tasks. #FLOPs denotes the average computational cost of KD for each instance. <sup>†</sup> denotes results from Jiao et al. (2020).

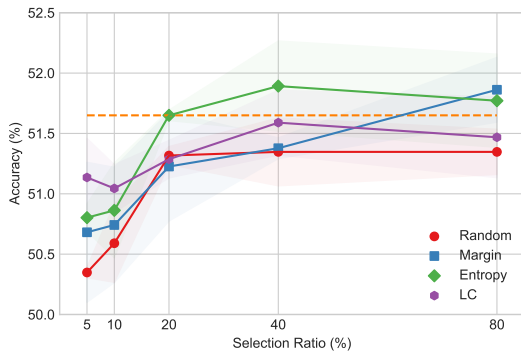


Figure 3: We plot the mean accuracy on the validation set of 3 seeds ( $\pm$  one standard deviation) under different selection ratios of various strategies. Orange dashed line denotes the performance of vanilla KD.

space, thus selecting the informative instances will not lead to significant performance drop. Besides, it is of great practical value to accelerate the KD procedure via reducing the queries to teacher model on the augmented dataset. For example, it costs about \$3,709 for querying all the instances of the augmented MNLI dataset as mentioned by Krishna et al. (2019).<sup>3</sup> By only querying a small portion of instances to the teacher model, we can greatly reduce economic cost and ease the possible environmental side-effects (Strubell et al., 2019; Schwartz et al., 2019; Xu et al., 2021) that may hinder the deployments of PLMs on downstream tasks.

The results with TinyBERT-4L as the backbone model and  $r = 0.1$  are listed in Table 4. We can observe that uncertainty-based selection strategy can maintain the superior performance while saving the computational cost, e.g., the FLOPs is reduced from 24.9B to 4.65B with negligible average performance decrease. In tasks like SST-5 and IMDB, selecting 10% most informative instances according to student prediction entropy can

<sup>3</sup>The cost is estimated according to Google Cloud natural language API: <https://cloud.google.com/natural-language/pricing>.

even outperform the original TinyBERT using the whole dataset. Among these strategies, the least-confidence strategy performs relatively poor, as it only takes the maximum probability into consideration while neglects the full output distribution.

**Performance under Different Ratios** We vary the selection ratio  $r$  to check the results of different strategies on the augmented SST-5 dataset. The results are shown in Figure 3. Our observations are: (1) There exists a trade-off between the performance and the training costs, i.e., increasing the selection ratio generally improves the performance of student model, while results in bigger training costs. (2) We can achieve the full performance using about 20% training data. It indicates that the training data support can be well covered with about 20% data, thus learning from these instances can sufficiently train the student model. It validates our motivation to select informative instances for reducing the repetitive learning caused by data redundancy. (3) Selection strategies based on the uncertainty of student prediction can make the better use of limited query, performing better than the random selection, especially when the query number is low.

### 3.2.3 Analysis

We further conduct experiments on the augmented SST-5 dataset to gain insights about the property of selected instances and visualize the distribution of selected instances for intuitive understanding.

**Property of Selected Instances** We plot the teacher prediction entropy and the distance from the selected instances to the corresponding category center. From Figure 4 (left), we observe for hard instances with high uncertainty that selected by the student model, the teacher model also regards them as difficult. It indicates that the instance difficulty is an inherent property of data and uncertainty-

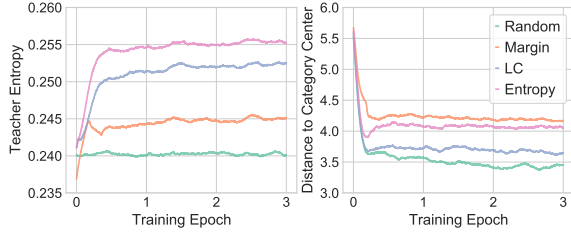


Figure 4: Training dynamics comparison of the selected instances. The entropy strategy can distinguish more informative instances from the whole dataset.

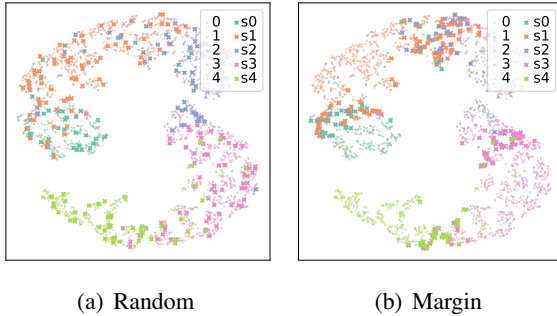


Figure 5: The t-SNE visualization of instance representations. Uncertainty-based strategies select the instances close to the class boundary, which is useful for the learning of the student model. Best viewed in color.

based criterion can discover these hard instances from the whole dataset. Besides, the teacher’s entropy of selected instances increases as the training proceeds, showing that the selected instances also evolve during the training as the student is becoming stronger. The right part in Figure 4 demonstrates that uncertainty-based selection will pick up the instances that are far away from the category center than the ones are randomly picked. These instances are more informative for the student model to learn the decision boundary of different classes.

**Visualization of Selected Instances** We further visualize the distribution of instances in the feature space, i.e., the representation before the classifier layer, using t-SNE (Maaten and Hinton, 2008) and highlight the selected instances with the cross marker. We compare the best performing strategy margin and random selection on SST-5. As shown in Figure 5, the instances randomly selected are distributed uniformly in feature space. The margin strategy instead picks the instances close to the classification boundary. The results demonstrate that the uncertainty-based selection criterion can help the student model pay more attention to the instances that are vital for making correct predictions,

thus achieving a comparable performance with a much lower computational cost.

In all, our analysis experiments show that the uncertainty-based selection is effective for picking instances that are close to the classification boundary. Besides, the selected instances also evolve as the student model becomes stronger. By learning from these instances, the computational cost of KD is greatly reduced with a negligible accuracy drop.

### 3.3 Dynamic Supervision Adjustment

We finally explore the question of the optimal learning objective functions. Previous studies have shown that integrating the alignments on the intermediate representation (Romero et al., 2015; Sanh et al., 2019; Sun et al., 2019) and attention map (Jiao et al., 2020; Wang et al., 2020) between the student and the teacher model can further boost the performance. We are interested in whether the dynamic adjustment of the supervision from different alignment objectives can bring extra benefits. As the first exploration, we only consider the combination of the KL-divergence distance with the teacher prediction and the hidden representation alignments:

$$\mathcal{L}_{KD} = \lambda_{KL} * \mathcal{L}_{KL} + \lambda_{PT} * \mathcal{L}_{PT} \quad (11)$$

where  $\mathcal{L}_{PT}$  is called PaTient loss, which measures the alignment between the normalized internal representations of the teacher and student model (Sun et al., 2019):

$$\mathcal{L}_{PT} = \sum_{i=1}^M \left\| \frac{\mathbf{h}_i^s}{\|\mathbf{h}_i^s\|_2} - \frac{\mathbf{h}_{I_{pt}(j)}^t}{\|\mathbf{h}_{I_{pt}(j)}^t\|_2} \right\|_2^2 \quad (12)$$

where  $M$  is the number of student layer,  $I_{pt}(i)$  denotes the corresponding alignment of the teacher layer for the student  $i$ -th layer,  $\mathbf{h}_i^s$  and  $\mathbf{h}_i^t$  denote representation of  $i$ -th layer of student and teacher model, respectively.

#### 3.3.1 Uncertainty-based Supervision Adjustment

Different from previous studies which set the corresponding alignment objective weights via hyperparameter search and keep them unchanged during the training, we propose to adjust the weights according to the student prediction uncertainty for each instance. The motivation behind is that we assume it is unnecessary to force the student model

Method	SST-5	MRPC	RTE	Avg.
BERT <sub>BASE</sub> (Teacher)	52.0	86.8	67.8	68.9
Vanilla KD	47.4	80.2	64.9	64.2
BERT-PKD	46.6	80.8	65.1	64.2
Uncertainty	<b>48.1</b>	<b>81.5*</b>	<b>66.4*</b>	<b>65.3</b>

Table 5: Results of dynamic adjusting the supervision weights, showing the uncertainty-based adjustment is effective. \* denote results are statistically significant with  $p < 0.05$ .

to align all the outputs of the teacher model during the whole training stage. As the training proceeds, the student is become stronger and it may learn the informative features different from the teacher model. Therefore, there is no need to force the student to act exactly with the teacher model, i.e., requiring the intermediate representations of the student to be identical with the teacher. Formally, we turn the weight of KD objective into a function of the student prediction uncertainty  $u(x) = \text{Entropy}(\sigma(S(s)))$ :

$$\lambda_{KL} = \lambda_{KL}^* (1 - \frac{u_x}{U}), \quad \lambda_{PT} = \lambda_{PT}^* \frac{u_x}{U} \quad (13)$$

where  $\lambda_{KL}^*$  and  $\lambda_{PT}^*$  are pre-defined weight for each objective obtained by parameter search and  $U$  is the normalization factor. In this way, the contribution of these two objectives are adjusted dynamically during the training for each instances. For instances that the student is confident about, the supervision from the internal representation alignment is down-weighted. Thus the student is focusing mimicking the final prediction probability distribution with the teacher based on its own understanding of the instance. On the contrary, for instances that the student is confusing, the supervision from teacher model representations can help it learn the feature of the instance better.

### 3.3.2 Experiments

**Settings** The student model is set to 6-layer and BERT<sub>BASE</sub> is adopted as the teacher model. For intermediate layer representation alignment, we adopt the Skip strategy, i.e.,  $I_{pt} = \{2, 4, 6, 8, 10\}$  as it performs best as described in BERT-PKD. We conduct experiments on the sentiment analysis task SST-5, and two natural language inference tasks MRPC and RTE. For  $\lambda_{KL}^*$  and  $\lambda_{PT}^*$ , we adopt the searched parameters provided by Sun et al. (2019).

**Results** The results of adaptive adjusting the supervision weights are listed in Table 5. We ob-

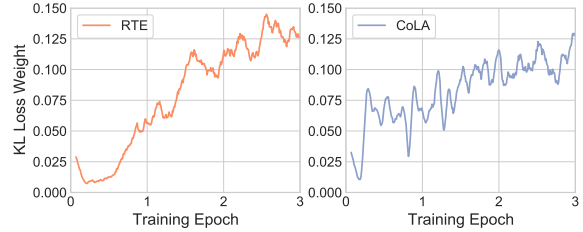


Figure 6: Evolution of the dynamically adjusted weight of KL-divergence loss weight.

serve that the proposed uncertainty-based supervision adjustment can outperform the static version BERT-PKD on all the tasks, showing that the proper adjustment of the KD objectives is effective for improving the student performance. We also plot the batch average of the KL loss weight in Figure 6. As expected, the corresponding weight of the prediction probability alignment objective is increasing as the student becomes more confident about its predictions, thus paying more attention to matching the output distribution with the teacher model. Interestingly, we find that at the initial stage of training, the KL weight is decreasing. It indicates that the learning by aligning the intermediate representations can help the student quickly gain the understanding the task, thus improving the confidence of predictions.

## 4 Discussions

After the preliminary explorations on the three aspects of Dynamic KD, we observe that it is promising for improving the efficiency and the distilled student performance. Here we provide potential directions for further investigations.

(1) From uncertainty-based selection criterion to advanced methods. In this paper, we utilize student prediction uncertainty as a proxy for selecting teachers, training instances and supervision objectives. More advanced methods based on more accurate uncertainty estimations (Gal and Ghahramani, 2016; Zhou et al., 2020), clues from training dynamics (Toneva et al., 2018), or even a learnable selector can be developed.

(2) From isolation to integration. As a preliminary study, we only investigate the three dimensions independently. Future work can adjust these components simultaneously and investigate the underlying correlation between these three dimensions for a better efficiency-performance trade-off.

(3) More fine-grained investigations regarding different components in the Dynamic KD frame-



work: (i) For teacher adoption, exploring whether dynamically training a student model from more teacher models or teacher models with different architectures can bring extra benefits; (ii) For data selection, it will be interesting to investigate whether the informative data is model-agnostic, and whether dynamically selecting data from different domains can improve the generalization performance; (iii) For supervision adjustment, investigations on the effect of combinations of different objectives can be promising.

## 5 Related Work

Our work relates to recent explorations on applying KD for compressing the PLMs and active learning.

**Knowledge Distillation for PLMs** Knowledge distillation (Hinton et al., 2015) aims to transfer the dark knowledge from a large teacher model to a compact student model, which has been proved effective for obtaining compact variants of PLMs. Those methods can be divided into general distillation (Sanh et al., 2019; Turc et al., 2019; Wang et al., 2020) and task-specific distillation (Sun et al., 2019; Jiao et al., 2020; Xu et al., 2020; Li et al., 2020a; Liang et al., 2021; Li et al., 2020b; Wu et al., 2021). The former conducts KD on the general text corpus while the latter trains the student model on the task-specific datasets. In this paper, we focus on the latter one as it is more widely adopted in practice. Compared to existing static KD work, we are the first to explore the idea of Dynamic KD, making it more flexible, efficient and effective.

**Active Learning** (Settles, 2009), where a learning system is allowed to choose the data from which it learns from, for achieving better performance with fewer labeled data. Traditional selection strategies include uncertainty-based methods (Scheffer et al., 2001; Settles, 2009), which select the instances that model is most uncertain about, query-by-committee (Freund et al., 1997), which select instances with highest disagreements between a set of classifiers, and methods based on decision theory (Roy and McCallum, 2001). In this paper, inspired by the success of active learning, we introduce Dynamic KD that utilizes the different strategies like prediction entropy as a proxy of student competency to adaptively adjust the different aspects of KD. Our explorations show that the uncertainty-based strategies are effective for improving the efficiency and performance of KD.

## 6 Conclusion

In this paper, we introduce dynamic knowledge distillation, and conduct exploratory experiments regarding teacher model adoption, data selection and the supervision adjustment. Our experimental results demonstrate that the dynamical adjustments on the three aspects according to the student uncertainty is promising for improving the student performance and learning efficiency. We provide discussions on the potential directions worth exploring in the future, and hope this work can motivate studies towards more environmental-friendly knowledge distillation methods.

## Acknowledgements

We thank all the anonymous reviewers for their constructive comments and Xuancheng Ren for his valuable suggestions in preparing the manuscript. This work was supported by a Tencent Research Grant. Xu Sun is the corresponding author of this paper.

## References

- Luisa Bentivogli, Ido Kalman Dagan, Dang Hoa, Danilo Giampiccolo, and Bernardo Magnini. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC Workshop*.
- Shrey Desai and Greg Durrett. 2020. *Calibration of pre-trained transformers*. In *EMNLP*, pages 295–302, Online.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. *BERT: Pre-training of deep bidirectional transformers for language understanding*. In *NAACL-HLT*, pages 4171–4186.
- William B. Dolan and Chris Brockett. 2005. *Automatically constructing a corpus of sentential paraphrases*. In *Proceedings of the Third International Workshop on Paraphrasing (IWP)*.
- Yoav Freund, H Sebastian Seung, Eli Shamir, and Naf-tali Tishby. 1997. Selective sampling using the query by committee algorithm. *Machine learning*, 28(2):133–168.
- Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, pages 1050–1059.
- Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *ICML*, pages 1321–1330.
- Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

- 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 EMNLP*, pages 4163–4174.
- Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. 2019. Thieves on sesame street! Model extraction of BERT-based APIs. In *ICLR*.
- Jianquan Li, Xiaokang Liu, Honghong Zhao, Ruifeng Xu, Min Yang, and Yaohong Jin. 2020a. [BERT-EMD: Many-to-many layer mapping for BERT compression with earth mover’s distance](#). In *EMNLP*, pages 3009–3018.
- Lei Li, Yankai Lin, Shuhuai Ren, Deli Chen, Xuancheng Ren, Peng Li, Jie Zhou, and Xu Sun. 2020b. Accelerating pre-trained language models via calibrated cascade. *arXiv preprint arXiv:2012.14682*.
- Kevin J. Liang, Weituo Hao, Dinghan Shen, Yufan Zhou, Weizhu Chen, Changyou Chen, and Lawrence Carin. 2021. MixKD: Towards efficient distillation of large-scale language models. In *ICLR*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *ICLR*.
- Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. [Learning word vectors for sentiment analysis](#). In *ACL*, pages 142–150.
- Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *JMLR*, 9:2579–2605.
- Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *AAAI*, pages 5191–5198.
- Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for thin deep nets. In *ICLR*.
- Nicholas Roy and Andrew McCallum. 2001. Toward optimal active learning through sampling estimation of error reduction. In *ICML*, pages 441–448.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *NeurIPS Workshop on Energy Efficient Machine Learning and Cognitive Computing*.
- Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pages 309–318. Springer.
- Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2019. Green AI. *arXiv preprint arXiv:1907.10597*.
- Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. [Recursive deep models for semantic compositionality over a sentiment treebank](#). In *EMNLP*, pages 1631–1642.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. [Energy and policy considerations for deep learning in NLP](#). In *ACL*, pages 3645–3650.
- Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. [Patient knowledge distillation for BERT model compression](#). In *EMNLP-IJCNLP*, pages 4323–4332.
- Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. 2018. An empirical study of example forgetting during deep neural network learning. In *ICLR*.
- Iulia Turc, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Well-read students learn better: The impact of student initialization on knowledge distillation. *arXiv preprint arXiv:1908.08962*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*, pages 5998–6008.
- Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *NeurIPS*.
- Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. [Neural network acceptability judgments](#). *TACL*, 7:625–641.
- Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. [A broad-coverage challenge corpus for sentence understanding through inference](#). In *NAACL-HLT*, pages 1112–1122.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. [Transformers: State-of-the-art natural language processing](#). In *System Demonstrations, EMNLP*, pages 38–45.

Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. 2021. One teacher is enough? pre-trained language model distillation from multiple teachers. In *Findings of ACL-IJCNLP 2021*, pages 4408–4413.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Furu Wei, and Ming Zhou. 2020. [BERT-of-Theseus: Compressing BERT by progressive module replacing](#). In *EMNLP*, pages 7859–7869.

Jingjing Xu, Liang Zhao, Junyang Lin, Rundong Gao, Xu Sun, and Hongxia Yang. 2021. [KNAS: green neural architecture search](#). In *ICML*, pages 11613–11625.

Yikai Zhou, Baosong Yang, Derek F Wong, Yu Wan, and Lidia S Chao. 2020. [Uncertainty-aware curriculum learning for neural machine translation](#). In *ACL*, pages 6934–6944.

## A Teacher Size Exploration Settings

We conduct the knowledge distillation with  $BERT_{BASE}$  and  $BERT_{LARGE}$  as teacher models. The student model is set to a 6-layer student BERT. For training the teacher, the teacher model is fine-tuned using the script provided by Huggingface Transformers library. The fine-tuning learning rate is  $2e-5$  with a linear warm-up learning rate schedule for the first 10% training steps. Batch size is 32, training epoch is set to 3, and the max length of input sentence is set to 128. The statistics of used datasets are listed in Table 3.

For distilling the student model, the student model is initialized using the first 6 layers weights of  $BERT_{BASE}$ . We adopt the KL-divergence distance as the KD objective.  $\lambda_{KL}$  is set to 0.5 and we empirically find this setting works well. The same training hyper-parameters as fine-tuning the teacher model are used for distillation. The performance is evaluated on the validation set and averaged on 3 random seeds.

## B Impacts of Prediction Smoothness

To examine the influence of less-softened teacher predictions, we conduct distillation experiments with various temperature  $\tau$  using the hyper-parameters identical with previous experiments, to mimicking the sharpen impact introduced by the larger teacher size. In more detail, we setup a student model with 6 layers as before and select the  $BERT_{BASE}$  as the teacher model. The results are illustrated in Figure 7. We observe on both datasets, the decreased temperature  $\tau$  will lead a performance decrease. It indicates that the less-softened probability distribution indeed weakens the performance of knowledge distillation.

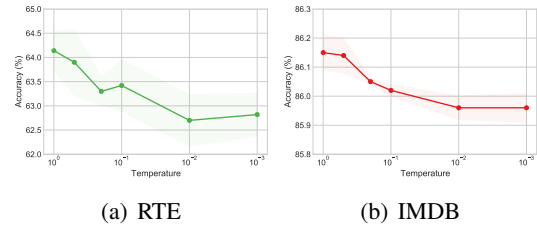


Figure 7: Distillation performance with varying temperature  $\tau$  on different datasets.

## C Impacts of Teacher Hidden Size

As mentioned in the main paper, we observe that larger teacher may not raise a student model with better performance. We further conduct experiments regarding the hidden size of teacher model. Specifically, we setup a student with 6 layers with 256 hidden units. The small teacher and the large teacher are a BERT model of 12-layer with 256 hidden units and a BERT of 12-layer with 768 hidden units, respectively. Our experiments show that on the CoLA dataset, the student model distilled with the small teacher can achieve 11.9 matthews correlation score while that of model distilled by the large teacher is 8.8. The result on the IMDB is consistent, i.e., 83.2 accuracy for student model distilled by the large teacher and 83.4 accuracy for the student distilled by the small teacher. These results again verify the phenomenon that the larger teacher may not always raise a better student model.

## D Implementation Details

Our implementation is based on PyTorch and Huggingface transformers library. Model is optimized with AdamW optimizer with linear learning rate warm-up. The sentence length is set to 64 for SST-5 and 128 for the rest datasets. Our teacher model is  $BERT_{BASE}$ . The model is trained with learning rate  $2e-5$  and batch size 32 for 3 epochs.  $\lambda_{KL}$  is set to 0.5, with temperature  $\tau$  set to 1.

For experiments using TinyBERT, we select TinyBERT<sub>4</sub> v2 as our backbone model, and conduct the general distillation for 10 epochs on the augmented dataset. We further train the model for 3 epochs on the augmented dataset and choose learning rates from  $\{1e-5, 2e-5, 3e-5\}$  and batch sizes from  $\{16, 32\}$  based on the performance on the validation set. The performance are evaluated on the test sets and  $\tau$  is set to 1, following the practice of TinyBERT.