Intermediate Layer Distillation with the Reused Teacher Classifier: A Study on the Importance of the Classifier of Attention-based Models

Hang Zhang, Seyyed Hasan Mozafari, James J. Clark, Brett H. Meyer, Warren J. Gross
Department of Electrical and Computer Engineering, McGill University
{hang.zhang3, sh.mozafari}@mail.mcgill.ca,
{james.clark1, brett.meyer, warren.gross}@mcgill.ca

Abstract

Intermediate Layer Distillation (ILD) effectively compresses large-scale pre-trained language models (PLMs). Existing ILD methods underestimate the importance of utilizing the teacher's discriminative classifier and face challenges in establishing proper layer mappings. Therefore, we propose **ILD-RTC**, to show that a straightforward implementation of reusing the pre-trained teacher classifier improves student performance even with simple uniform layer mapping. Through extensive experiments, our method outperforms other ILD techniques, maintaining 97.7% performance of the original teacher BERT_{base} without additional trainable parameters. Projectors are developed to help the student match the hidden size of the teacher model, making our ILD-RTC applicable to students with different sizes. In addition, our technique achieves the same average GLUE score as students initialized by pre-trained LMs, saving over $80\times$ cost resulting from the pre-training step. Our method emphasizes the reuse of pre-trained teacher classifiers as an alternative to pre-training the student for initialization.

1 Introduction

PLMs, such as BERT (Devlin et al., 2019), LLaMA (Touvron et al., 2023) and GPT-4 (Achiam et al., 2023) show great success and achieve remarkable accuracy on various Natural Language Processing (NLP) tasks. However, to make PLMs adaptive to diverse tasks, they are always over-parameterized (Tahaei et al., 2022).

Knowledge Distillation (KD) (Hinton et al., 2015) is designed to help a less-parameterized model (student) gain a comparable performance as PLMs (teacher) while maintaining the generalization capability. Vanilla KD simply forces the student model to mimic the logits of the teacher. However, the teacher-student performance gap is still significant. In recent years, many approaches have been proposed to narrow this gap. For instance, some of them apply TA networks (Mirzadeh et al., 2019), balance the KL divergence (Amara et al., 2022), or use other distillation objectives (Tung and Mori, 2019). Students also benefit from additional supervision for the intermediate layers (Sun et al., 2019; Jiao et al., 2020; Wu et al., 2021). Despite their impressive results, most Intermediate Layer Distillation (ILD) methods are typically based on mimicking the hidden representatives and rely on careful hyperparameter fine-tuning (Chen et al., 2022). However, their success is not ensured and can hardly be reproduced in practice (Liang et al., 2023). Another limitation of ILD is that the layer mapping should be carefully determined, and there are no universal criteria for various model architectures and distillation objectives (Ko et al., 2023). The diversity of knowledge transferred in the KD process highlights an urgent need for an effective, straightforward, and unified solution that can be adaptive to diverse attention-based students.

In this paper, we propose Intermediate Layer Distillation with the Reused Teacher Classifier (**ILD-RTC**). By constructing a student network with the reused pre-trained teacher classifier, we empirically demonstrate that knowledge obtained from a large corpus is embedded not only in the hidden layers but also within the classifier. Our contributions can be summarized as follows:

- The reused classifier provides a better initialization for the student model and significantly bridges the performance gap between students and teachers (PLMs).
- Our technique is adaptive to attention-based students of different sizes. The idea of reusing can be flexibly used as an orthogonal approach in addition to various ILD methods or layer mapping strategies.
- ILD-RTC is simple to implement. The training time and the peak GPU memory usage

Findings of the Association for Computational Linguistics: EACL 2024, pages 7203–7212
November 12-16, 2024 ©2024 Association for Computational Linguistics

are almost the same as fine-tuning, without additional cost on extra layers or parameters.

2 Related Work

Different from vanilla KD (Hinton et al., 2015) which solely learns from the teacher's prediction, BERT-PKD (Sun et al., 2019) leverages more information from intermediate layers to extract knowledge in the teacher model. However, one of the biggest challenges lies in finding a proper layer mapping function. Literature (Passban et al., 2020) points out that when the number of hidden layers for the student network is smaller than that of the teacher, *Skip* strategy ignores certain layers in the ILD process so that the capability of the teacher network is not fully expressed. To solve this problem, pseudo classifiers are furnished to all the teacher and student layers in Universal-KD (Wang et al., 2020). They apply attention-based layer projection to find interpretability across the layers. In addition to incorporating more distillation objectives, TinyBERT (Jiao et al., 2020) deploys Data Augmentation (DA) to nearly match the teacher performance. TinyBERT's impressive performance comes at the expense of increased training time due to the additional pre-training stage and the larger training dataset required during the distillation process. Considering model compression aims to reduce training costs, existing solutions that need more complex architectures or training time are unsuitable.

Despite the investigation for intermediate features, literature (Chen et al., 2022) shows reusing the discriminative classifier of the pre-trained teacher for student inference benefits accuracy. They train a student with the same performance as the teacher model using standard KD settings. However, the importance of the pre-trained classifier is only studied in the context of Computer Vision (CV) tasks.

By replacing the randomly initialized student classifier with the pre-trained teacher classifier in KD solution, we improve student accuracy without adding implementation complexity or incurring additional training costs. The knowledge contained in the reused classifier also compensates for the missing information from skipped intermediate layers.

3 Methodology

Figure 1 shows an overview of our proposed methodology. The student is trained to minimize

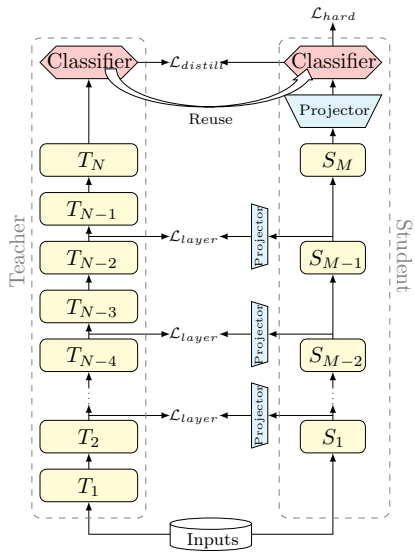

Figure 1: An overview of our proposed ILD-RTC. The yellow shaded blocks are intermediate layers. The embedding layer is omitted.

the loss function consisting of layerwise loss, distillation loss, and cross-entropy loss. The pre-trained teacher classifier is reused to construct the student network.

3.1 Intermediate Layer Distillation

To simplify the student learning procedure and save effort on layer mapping design, we apply uniform layer mapping in ILD. The symbolic description is illustrated as follows.

Assume that the student model has M transformer layers and the teacher model has N transformer layers. If we discard the embedding layer and set $[1:M]$ and $[1:N]$ to be the indices of the student layers and the teacher layers respectively, the mapping function from the student layers to the teacher layers will be $g(m) = \lfloor \frac{N}{M} \rfloor \times m$ when we apply the uniform strategy.

To maintain the generalization capability of the original teacher model, our students are designed to imitate the representatives of the first token in the intermediate layers, using the same manner as BERT-PKD (Sun et al., 2019). The layer mapping for the last layer of the student network is omitted since those features are counted in KD loss $\mathcal{L}_{distill}$ in Equation 3. Thus, for an input $\mathbf{x}_i$, the outputs of the first tokens for all the supervised student transformer layers are $\mathbf{h}_i = [\mathbf{h}_{i,1}, \mathbf{h}_{i,2}, \cdots, \mathbf{h}_{i,k}]$, where $k = M - 1$ is the index for the last supervised student layer. We denote the set of sampled teacher transformer layers where $\mathbf{h}_i$ distill from as $\mathcal{O}$. Suppose the teacher has 12 transformer layers,

then $\mathcal{O} = \{2, 4, 6, 8, 10\}$ if the student has 6 intermediate layers while $\mathcal{O} = \{3, 6, 9\}$ for a 4-layer student.

Considering the hidden size of the student model may be smaller than that of the teacher model, we apply projectors to map the hidden states of the student network into the same space as the teacher's. The layerwise distillation objective is defined as the mean-square error (MSE) between the normalized hidden states:

$$\mathcal{L}_{layer} = \sum_{i=1}^{P} \sum_{j=1}^{M-1} \left\| \frac{\mathbf{h}_{i,j}^{s}}{\left\|\mathbf{h}_{i,j}^{s}\right\|_2} - \frac{\mathbf{h}_{i,\mathcal{O}(j)}^{t}}{\left\|\mathbf{h}_{i,\mathcal{O}(j)}^{t}\right\|_2} \right\|_2^2 \tag{1}$$

where P denotes the number of training samples, $\mathbf{h}_i \in \mathbb{R}^{k \times d}$ indicates the outputs of the first token *[CLS]*, d is the hidden size, and the superscripts s and t for $\mathbf{h}$ represent the student and the teacher model, respectively.

3.2 Teacher Classifier Reusing

Unlike existing ILD techniques, we reuse the teacher classifier in the student model to fully leverage the knowledge gained by the teacher in the fine-tuning stage on the target dataset.

The overall distillation objective is defined as:

$$\mathcal{L} = \alpha \mathcal{L}_{hard} + (1 - \alpha)\mathcal{L}_{distill} + \beta \mathcal{L}_{layer} \tag{2}$$

where α is the hyperparameter that balances the cross-entropy loss w.r.t. the hard labels and the distillation loss compared with the teacher's soft logits. β is another hyperparameter that weights the importance of the features for distillation regarding the intermediate layers, explained in Section 3.1. $\mathcal{L}_{hard}$ and $\mathcal{L}_{distill}$ are specifically defined as:

$$\begin{aligned} \mathcal{L}_{hard} &= \mathrm{CE}(\mathbf{z}^s, \text{label}) \\ \mathcal{L}_{distill} &= \mathrm{KL}(\log(\sigma(\mathbf{z}^s/T)), \sigma(\mathbf{z}^t/T)) \end{aligned} \tag{3}$$

T is the temperature to control the shape of $softmax$ function σ, $\mathbf{z}^s$ and $\mathbf{z}^t$ are generated logits by the student model with the reused teacher classifier and the teacher model, respectively.

With the projectors shown in Figure 1, ILD-RTC adapts to attention-based students of various hidden sizes. Additionally, since language models typically have a discriminative classifier for sequence classification tasks, the idea of reusing the classifier can be flexibly combined with other ILD methods and is not constrained by model architectures.

4 Experiments and Results

4.1 Datasets

We evaluate our proposed methodology on six GLUE (Wang et al., 2018) downstream tasks. The Stanford Sentiment Treebank (SST-2) is a single-sentence task. Microsoft Research Paraphrase Corpus (MRPC) and Quora Question Pairs (QQP) are similarity and paraphrase tasks while Recognizing Textual Entailment (RTE), Questioning Natural Language Inference (QNLI), and Multi-Genre Natural Language Inference (MNLI) are the inference tasks.

4.2 Teacher Models and Student Networks

Our teacher models are standard BERT$_{base}$ fine-tuned on GLUE downstream tasks. There are 12 encoder layers and each of them contains 12 attention heads. The hidden dimension and the feed-forward dimension are 768 and 3072 respectively.

We first test our method ILD-RTC on shallower BERT variants. We construct two student models containing 6 layers and 4 layers correspondingly, while other configurations, including hidden dimensions, and feed-forward dimensions are kept the same as BERT$_{base}$. We utilize the first few layers from the pre-trained teacher to initialize our students, since information captured by these layers is more generic rather than task-specific, ensuring the generalization capability of compressed models. See Appendix E for further details.

To show that our proposed ILD-RTC supports student networks with different hidden sizes, we also validate our method on other BERT variants, further discussed in Appendix B. From the results, ILD-RTC is effective across various BERT-based students and enhances student accuracy.

4.3 Experimental Setup

Our student models only need a single-stage distillation of 4 epochs. The overall distillation objective is in Equation 2. During training for all six tasks, the batch size is fixed to 8 and the maximum sequence length is 128. The temperature T in Equation 3 is set as 10. We perform a grid search over other hyperparameters, including the learning rate from the set $\{1e-5, 2e-5, 5e-5\}$, $\alpha = \{0.2, 0.5, 0.7\}$, and $\beta = \{10, 100, 500, 1000\}$. The best hyperparameter values are listed in Appendix A.

	SST-2	MRPC	QQP	MNLI-m/mm	QNLI	RTE	Avg
Number of Samples	67,349	3,668	363,846	392,702	104,743	2,490	
BERT$_{base}$ (Teacher)	93.2	87.9	71.1	83.4/83.6	90.7	68.9	82.7
BERT-PKD$_6$	92.0	85.0	70.7	81.5/81.0	**89.0**	**65.5**	80.7
DistilBERT$_6$	**92.3**	**87.6**	69.6	81.6/81.3	88.8	54.1	79.3
ILD-RTC$_6$ (Ours)	92.0	85.4	**71.3**	**82.2/81.6**	88.0	65.0	**80.8**
BERT-PKD$_4$	**90.2**	82.1	69.2	79.1/78.5	86.0	61.4	78.1
DistilBERT$_4$	91.4	82.4	68.5	78.9/78.0	85.2	54.1	76.9
Universal-KD$_4$	**90.2**	**84.1**	68.9	79.3/78.9	**86.3**	62.6	**78.6**
ILD-RTC$_4$ (Ours)	89.8	82.7	**69.8**	**79.9/78.7**	84.7	**62.9**	78.3

Table 1: The comparison results from the GLUE test server. The results of DistilBERT$_6$ are obtained from the GLUE benchmark leaderboard. The results of BERT-PKD, DistilBERT$_4$, and Universal-KD$_4$ are from (Sun et al., 2019; Jiao et al., 2020; Wu et al., 2021). Our ILD-RTC test results are from students trained with the best-performing hyperparameters for each task. Student models in the same group have identical numbers of hidden layers, hidden dimensions, and feed-forward dimensions.

	SST-2	MRPC	QQP	MNLI-m/mm	QNLI	RTE	Avg
BERT$_6$	90.4	87.3	87.1	81.5/81.5	87.1	65.7	82.9
+ ILD	89.9	88.3	86.8	81.1/82.1	87.1	63.2	82.6
+ RTC (Ours)	**91.2**	**89.8**	**88.2**	**82.7/83.1**	**88.3**	**67.1**	**84.3**
BERT$_4$	88.4	84.8	85.9	78.4/78.4	85.4	63.2	80.6
+ ILD	88.1	85.6	86.0	78.8/79.3	85.9	60.3	80.6
+ RTC (Ours)	**90.1**	**86.8**	**87.1**	**80.4/80.6**	**86.2**	**63.9**	**82.2**
BERT-PD$_6$	91.1	89.4	87.4	82.5/**83.4**	**89.4**	66.7	84.3
ILD-RTC (Ours)	**91.2**	**89.8**	**88.2**	**82.7**/83.1	88.3	**67.1**	**84.3**

Table 2: The effectiveness of reusing the teacher classifier, on dev sets.

4.4 Student Accuracy and Training Time

The comparison results on the six datasets are summarized in Table 1. The first group is the comparison between 6-layer students, while the second group is for 4-layer students. Our student ILD-RTC$_6$ has 67M parameters, while ILD-RTC$_4$ has a total of 52M parameters. Overall, our proposed ILD-RTC obtains a higher average score than both BERT-PKD and DistilBERT. Notably, BERT-PKD is a widely recognized baseline in ILD, while DistilBERT applies vanilla KD approach but with a better initialization. In addition, our students get the best accuracy on QQP and MNLI among all students, illustrating that students benefit more from the pre-trained teacher classifier when the tasks are more complex.

We also conduct experiments on CoLA, one of the downstream tasks of GLUE benchmark, detailed results are discussed in Appendix D. Including these results, our model outperforms Universal-KD. Furthermore, our method achieves high-quality results with a single-stage distillation running for 4 epochs, whereas Universal-KD requires a two-stage process, with the first stage involving 20 epochs of distillation and the second stage containing 4 epochs of training. Moreover, we do not need extra pseudo classifiers. Therefore, our technique is straightforward to implement and training goes more quickly.

We focus on task-specific distillation and optimize the distillation objective to make students converge faster. The total training time for all the reported six downstream tasks is 9 hours for the 6-layer student and 7 hours for the 4-layer student on a single NVIDIA V100 GPU, which is dramatically reduced compared to DistilBERT (720 GPU hours) (Sanh et al., 2020) and TinyBERT (576 GPU hours) (Wang et al., 2023) on the same platform. The training time and the peak memory usage are almost the same as fine-tuning.

4.5 The Effectiveness of Reusing the Classifier

We show the effectiveness of our proposed ILD-RTC by conducting experiments using different distillation objectives. The results are shown in Table 2.

We report the performance of both the 6-layer student and the 4-layer student. The baselines for the first two groups are fine-tuned BERT models.

ILD refers to the naive Intermediate Layer Distillation. RTC results are from our proposed ILD-RTC. Compared with ILD, our proposed method significantly narrows the teacher-student performance gap.

We also compare our students with the distilled students initialized by pre-trained masked LMs (MLMs), shown in the third group of Table 2. BERT-PD baseline and its results are from Google open source (Turc et al., 2019). Our student achieves a slightly higher average score than BERT-PD and outperforms it in five out of six downstream tasks (excluding MNLI-mm). Given that pre-training a student offers an initialization for distillation, and considering our method performs better than BERT-PD, utilizing the pre-trained teacher classifier for the compressed model proves more efficient than pre-training the entire student network. ILD-RTC enhances student performance efficiently and cost-effectively in practice.

The results indicate that knowledge in PLMs resides not only in the hidden layers but also in the classifier, emphasizing the importance of the discriminative classifier of PLMs and the effectiveness of reusing the pre-trained classifier in KD solutions.

5 Conclusion

In this paper, we present **ILD-RTC**, a method that employs intermediate layer distillation and improves student performance by directly deploying the teacher classifier. Experimental results on the GLUE benchmark show that the classifier contains transferable knowledge and therefore is of great importance. Reusing the pre-trained classifier reduces the teacher-student performance gap. Moreover, utilizing the teacher classifier can be an efficient and effective initialization for compact models, compared to pre-training the entire model resulting in expensive resource and memory costs.

Limitations

In our proposed method, our students are initialized by the lower layers of the BERT_{base} model. However, a better initialization may exist. Therefore, we plan to deploy different initializations for the hidden layers of the student model. Besides, in this paperwork, we mainly study the effectiveness of reusing the pre-trained classifier on BERT variants. Considering the flexible nature of adding the classifier on top of any sequence classification model, we can explore diverse student architectures and validate our proposed ILD-RTC in the field of cross-architecture KD, for instance, from attention-based models to BiLSTMs or TextCNNs.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ibtihel Amara, Nazanin Sepahvand, Brett H. Meyer, Warren J. Gross, and James J. Clark. 2022. Bd-kd: Balancing the divergences for online knowledge distillation. *Preprint*, arXiv:2212.12965.

Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. 2022. Knowledge distillation with the reused teacher classifier. *Preprint*, arXiv:2203.14001.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *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 the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Jongwoo Ko, Seungjoon Park, Minchan Jeong, Sukjin Hong, Euijai Ahn, Du-Seong Chang, and Se-Young Yun. 2023. Revisiting intermediate layer distillation for compressing language models: An overfitting perspective. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 158–175, Dubrovnik, Croatia. Association for Computational Linguistics.

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. 2023. Less is more: Task-aware layer-wise distillation for language model compression. *Preprint*, arXiv:2210.01351.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2019. Improved knowledge distillation via teacher assistant. *Preprint*, arXiv:1902.03393.

Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. 2020. Alp-kd: Attention-based layer projection for knowledge distillation. *Preprint*, arXiv:2012.14022.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

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

Marzieh Tahaei, Ella Charlaix, Vahid Nia, Ali Ghodsi, and Mehdi Rezagholizadeh. 2022. KroneckerBERT: Significant compression of pre-trained language models through kronecker decomposition and knowledge distillation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2116–2127, Seattle, United States. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Frederick Tung and Greg Mori. 2019. Similarity-preserving knowledge distillation. *Preprint*, arXiv:1907.09682.

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

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

Wenhui Wang, 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. *Preprint*, arXiv:2002.10957.

Xinpeng Wang, Leonie Weissweiler, Hinrich Schütze, and Barbara Plank. 2023. How to distill your BERT: An empirical study on the impact of weight initialisation and distillation objectives. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1843–1852, Toronto, Canada. Association for Computational Linguistics.

Yimeng Wu, Mehdi Rezagholizadeh, Abbas Ghaddar, Md Akmal Haidar, and Ali Ghodsi. 2021. Universal-KD: Attention-based output-grounded intermediate layer knowledge distillation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7649–7661, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A Hyperparameter Tuning

In Section 4.4, we report the accuracy of ILD-RTC$_6$ and ILD-RTC$_4$ from the GLUE test server. Both students are initialized by the first few layers of the pre-trained teacher model, BERT$_{base}$. The results are from trained students under the best-performing hyperparameter values. The corresponding hyperparameter values are listed in Table 3 and Table 4.

We follow the same training steps in distillation for all downstream tasks. The sets for hyperparameters are the same, which are $lr = \{1e-5, 2e-5, 5e-5\}$, $\alpha = \{0.2, 0.5, 0.7\}$, and $\beta = \{10, 100, 500, 1000\}$.

B Model Accuracy for Diverse Sizes

Our proposed method, ILD-RTC is adaptive to BERT variants with diverse sizes. It supports different hidden sizes and/or numbers of layers. To further validate the effectiveness of reusing the teacher classifier, we conduct experiments on other 4 attention-based students. The student performance and the corresponding model size are summarized in Table 5. Except for BERT$_{base}$ which is not a compressed model, ILD-RTC shows great capability to narrow the teacher-student performance gap. In addition, when the student's model size becomes smaller, the improvement through the reused pre-trained classifier is more significant.

C Contribution of Intermediate Layer Distillation

To verify the need for label information and extra gradient information from intermediate layers, we reproduced SimKD (Chen et al., 2022) on NLP tasks. The comparison between our proposed ILD-RTC and the reproduced work is shown in Table 6. We show improvement in accuracy by adding more gradient information.

D Discussions on CoLA Task

The Corpus of Linguistic Acceptability (CoLA) is one of the single-sentence tasks in the GLUE benchmark (Wang et al., 2018). Since some of

	SST-2	MRPC	QQP	MNLI	QNLI	RTE
α	0.2	0.5	0.5	0.2	0.7	0.5
β	10	500	500	100	1000	1000
learning rate	1e-5	5e-5	1e-5	1e-5	2e-5	5e-5

Table 3: Hyperparameter values for 6-layer student supervised by BERT$_{base}$.

	SST-2	MRPC	QQP	MNLI	QNLI	RTE
α	0.2	0.5	0.5	0.5	0.2	0.5
β	1000	500	10	1000	1000	10
learning rate	2e-5	1e-5	2e-5	2e-5	2e-5	1e-5

Table 4: Hyperparameter values for 4-layer student supervised by BERT$_{base}$.

the compressed models that we compared with do not report the performance on CoLA, we eliminate CoLA in Table 1 for a fair comparison. The detailed results on the dev set and the test set are shown in Table 7. In addition, we observe that Matthew's Correlation used for CoLA evaluation has higher variability in scores. The score is sensitive to model sizes and hyperparameters.

E Layerwise KL Divergence Analysis

In Figure 2, we show the KL divergence between each layer of the fine-tuned BERT$_{base}$. These heat maps illustrate that each layer of BERT$_{base}$ captures different representatives for the input word tokens. Moreover, the lower layers learn general information, while the higher layers are more task-specific. To make our students maintain the generalization capability, we initialize the students using the lower layers of the teacher model BERT$_{base}$.

F Further Validation on the Effectiveness of the Reused Teacher Classifier

To show the importance of the pre-trained teacher classifier, we run experiments using different distillation objectives. From the results in Table 8, we show that reusing the teacher classifier improves student accuracy under different conditions. In addition, when solely using layerwise distillation loss, the student's convergence is much slower. Reusing the teacher classifier for inference improves student accuracy and helps the student converge.

G Layer Mapping Discussion

Except the uniform layer mapping mentioned earlier in Section 3.1, we also consider other layer mapping strategies. As discussed in Appendix E, the top layers contain task-specific knowledge, therefore, we map the student layers to the last few layers of the teacher model. However, when comparing results (shown in Table 9) using these two layer mapping strategies, we do not observe much difference and both of them achieve comparable scores with baselines using complex mapping functions. Therefore, we save effort on finding proper layer mapping.

		SST-2	MRPC	QQP	MNLI-m/mm	QNLI	RTE	Avg
Fine-tuning	**BERT$_{base}$**	92.4	89.1	87.9	84.4/84.9	91.1	64.7	84.9
	BERT$_{medium}$	90.0	85.3	85.7	80.3/80.8	88.8	61.0	81.7
	BERT$_{small}$	87.0	82.5	84.3	77.5/77.7	86.6	59.6	79.3
	BERT$_{mini}$	85.4	82.9	81.2	73.4/74.8	83.8	60.6	77.4
	BERT$_{tiny}$	80.4	81.2	76.4	65.2/66.7	77.1	60.6	72.5
ILD-RTC	**BERT$_{base}$**	91.4	90.2	87.3	83.7/84.1	90.7	64.3	84.5
	BERT$_{medium}$	89.6	88.7	86.1	81.0/81.9	88.4	65.7	83.1
	BERT$_{small}$	88.9	88.4	85.7	78.7/79.0	86.5	65.0	81.7
	BERT$_{mini}$	86.2	86.6	83.6	76.4/77.5	84.0	58.8	79.0
	BERT$_{tiny}$	82.7	82.4	80.8	70.5/71.3	77.3	56.7	74.5

Table 5: ILD-RTC benefits various students, and results are reported on dev sets. Note: All these BERT variants are initialized by Google pre-trained models. The numbers of parameters for BERT$_{base}$, BERT$_{medium}$, BERT$_{small}$, BERT$_{mini}$ and BERT$_{tiny}$ are 110.1M, 41.7M, 29.1M, 11.3M, and 4.4M, respectively. The dimensions correspondingly are 12/768, 8/512, 4/512, 4/256 and 2/128, where the first number in each pair is the number of transformer layers and the second one is the hidden size.

Model	**SST-2**	**MRPC**	**QQP**	**MNLI-m/mm**	**QNLI**	**RTE**	**Avg**
BERT$_{base}$ (Teacher)	92.4	89.1	87.9	84.4/84.9	91.1	64.7	84.9
SimKD$_6$ (Reproduced)	90.4	87.7	86.6	81.4/81.5	87.8	65.0	82.9
ILD-RTC$_6$ (Ours)	**91.2**	**89.9**	**88.2**	**82.7/83.1**	**88.3**	**67.1**	**84.3**

Table 6: Comparison between directly reusing and reusing with extra gradient information from intermediate layers.

Model Name	**Score**	
	(dev)	(test)
BERT$_{base}$ (Teacher)	52.1	/
DistilBERT$_6$	51.3	/
BERT-PKD$_6$	37.4	/
ILD-RTC$_6$ (Ours)	44.8	33.4
BERT-PKD$_4$	25.1	25.4
Universal-KD$_4$	34.2	27.0
ILD-RTC$_4$ (Ours)	35.9	30.5

Table 7: CoLA results comparison, on both dev set and test set. Results for CoLA datasets have not been reported in some previous work, so we leave them blank in the table.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.00	0.06	0.05	0.55	0.46	6.25	8.01	9.08	7.30	4.67	4.22	18.70
2	0.06	0.00	0.00	0.32	0.28	7.09	8.95	10.16	8.23	5.16	4.41	19.31
3	0.05	0.00	0.00	0.33	0.28	6.97	8.80	10.01	8.11	5.11	4.33	19.24
4	0.57	0.33	0.34	0.00	0.09	9.93	12.11	13.74	11.45	7.36	5.59	21.94
5	0.47	0.29	0.29	0.09	0.00	8.76	10.84	12.55	10.36	6.55	4.94	20.42
6	6.02	6.79	6.67	9.31	8.28	0.00	0.25	0.73	0.53	1.31	6.37	9.34
7	7.76	8.61	8.47	11.40	10.29	0.25	0.00	0.40	0.42	1.63	4.55	9.28
8	8.78	9.76	9.63	12.92	11.88	0.73	0.40	0.00	0.32	2.21	5.67	10.28
9	7.06	7.90	7.80	10.76	9.81	0.53	0.41	0.32	0.00	1.31	4.24	9.63
10	4.54	4.98	4.94	6.91	6.19	1.36	1.67	2.29	1.36	0.00	1.39	8.18
11	4.27	4.40	4.34	5.33	4.76	4.04	4.98	6.21	4.66	1.48	0.00	9.21
12	20.08	20.49	20.44	22.45	21.07	11.04	10.97	12.40	11.45	9.18	9.86	0.00

(a) Layerwise KL divergence ($\times 10^{-2}$) on RTE.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.00	0.07	0.05	0.06	0.47	13.32	13.13	14.35	6.57	5.24	17.38	28.63
2	0.07	0.00	0.01	0.06	0.73	14.70	14.48	15.73	7.39	5.23	16.61	27.92
3	0.05	0.01	0.00	0.05	0.64	14.27	14.05	15.25	7.07	5.13	16.72	27.98
4	0.06	0.06	0.05	0.00	0.53	13.89	13.75	14.99	7.03	5.51	17.52	29.05
5	0.46	0.72	0.63	0.52	0.00	9.50	9.45	10.66	4.59	6.23	21.01	31.90
6	13.85	15.22	14.80	14.41	9.92	0.00	0.23	0.55	3.44	18.92	45.82	52.71
7	13.60	14.93	14.51	14.20	9.84	0.23	0.00	0.35	2.86	17.44	43.79	49.85
8	14.85	16.21	15.75	15.48	11.09	0.56	0.35	0.00	2.62	17.36	43.38	48.36
9	6.63	7.42	7.11	7.07	4.66	3.26	2.76	2.52	0.00	6.87	26.44	32.31
10	5.57	5.52	5.43	5.81	6.72	19.48	18.12	18.04	7.45	0.00	8.03	14.37
11	26.08	24.62	24.88	26.03	32.11	65.77	63.19	62.65	40.08	11.38	0.00	4.47
12	46.88	44.97	45.27	47.00	53.95	89.43	85.44	83.46	57.92	23.21	5.04	0.00

(b) Layerwise KL divergence ($\times 10^{-2}$) on MRPC.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.0	0.9	3.0	3.8	2.3	9.2	11.6	14.8	18.2	18.8	24.0	28.4
2	0.9	0.0	1.6	2.5	1.4	7.9	9.6	11.5	14.8	17.5	23.9	33.5
3	3.0	1.7	0.0	1.8	1.5	11.5	10.9	8.4	12.0	16.6	21.9	42.6
4	3.9	2.6	1.8	0.0	0.9	9.2	8.8	8.1	12.8	18.2	27.0	42.7
5	2.3	1.4	1.5	0.9	0.0	8.7	8.5	9.4	14.7	18.2	26.9	38.6
6	9.1	7.7	11.2	8.9	8.5	0.0	5.1	15.1	20.1	25.8	36.9	34.7
7	11.6	9.5	10.6	8.7	8.5	5.2	0.0	7.7	14.5	18.3	31.4	42.1
8	15.7	12.2	8.7	8.5	10.0	15.9	8.0	0.0	4.3	15.3	25.3	64.3
9	19.6	15.9	12.6	13.5	15.7	21.5	15.3	4.3	0.0	14.3	21.4	66.7
10	20.5	18.9	17.3	19.3	19.2	27.9	19.2	15.3	14.0	0.0	10.3	42.6
11	26.2	26.0	23.1	29.0	28.9	41.4	35.0	26.5	21.9	10.7	0.0	41.2
12	27.8	32.7	41.1	41.4	37.9	34.4	41.5	60.8	62.1	39.8	38.4	0.0

(c) Layerwise KL divergence ($\times 10^{-2}$) on SST-2.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.00	0.21	0.15	1.96	7.18	24.76	22.65	23.33	19.18	17.74	35.75	61.59
2	0.21	0.00	0.16	1.27	5.82	22.45	20.73	21.75	17.67	18.14	35.38	61.56
3	0.15	0.16	0.00	1.60	6.62	24.01	22.02	22.71	18.65	17.67	35.27	61.20
4	1.93	1.26	1.58	0.00	2.10	15.53	14.38	16.09	12.59	17.69	34.52	59.45
5	7.26	5.94	6.70	2.14	0.00	7.93	7.60	10.41	7.92	18.34	34.27	57.53
6	31.12	28.67	30.40	20.33	10.32	0.00	0.73	2.98	3.10	20.51	30.95	44.73
7	28.52	26.48	27.93	18.84	9.76	0.72	0.00	1.34	1.54	16.33	28.22	41.39
8	29.18	27.47	28.51	20.60	12.87	3.06	1.40	0.00	0.51	11.91	23.93	36.41
9	22.45	20.88	21.93	15.07	9.18	3.00	1.50	0.47	0.00	11.27	24.58	38.75
10	23.32	24.31	23.34	23.59	23.59	22.70	17.74	12.35	12.56	0.00	11.93	25.81
11	54.96	54.89	54.22	53.63	52.19	41.10	37.91	31.47	33.52	16.62	0.00	16.15
12	183.75	185.40	182.63	179.09	170.48	122.15	114.23	97.35	106.76	71.59	33.72	0.00

(d) Layerwise KL divergence ($\times 10^{-2}$) on QNLI.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.00	0.14	0.31	2.25	8.00	19.94	13.54	15.10	23.93	33.19	49.64	57.57
2	0.14	0.00	0.30	1.93	7.31	18.69	13.08	14.54	23.52	33.02	49.36	57.47
3	0.31	0.30	0.00	2.02	7.87	20.09	12.76	13.77	21.63	30.23	46.41	53.51
4	2.33	2.00	2.08	0.00	2.78	12.05	6.62	7.86	16.41	26.56	40.75	50.79
5	8.91	8.15	8.70	3.01	0.00	4.36	3.05	5.00	14.90	27.16	37.23	49.91
6	25.82	24.16	25.83	15.33	5.33	0.00	6.64	9.40	22.17	39.08	45.18	62.90
7	16.58	16.05	15.49	7.82	3.44	6.08	0.00	1.06	8.14	19.43	28.25	41.75
8	19.13	18.52	17.25	9.63	5.94	9.17	1.09	0.00	4.20	14.04	23.61	36.49
9	39.36	39.16	35.35	26.57	22.87	28.22	10.89	5.48	0.00	4.58	14.56	26.41
10	88.71	89.04	80.65	69.93	66.83	76.70	42.14	29.20	7.14	0.00	7.59	17.86
11	153.67	153.65	142.05	126.11	115.69	122.82	80.49	63.20	29.95	12.09	0.00	8.30
12	236.97	237.79	220.05	207.65	199.89	216.49	153.65	129.95	81.22	47.64	16.35	0.00

(e) Layerwise KL divergence ($\times 10^{-2}$) on QQP.

	1	2	3	4	5	6	7	8	9	10	11	12
1	0.00	0.18	0.17	3.55	6.37	9.68	9.94	22.07	36.11	48.25	67.91	99.26
2	0.18	0.00	0.30	4.74	7.95	10.98	11.14	22.62	36.32	48.28	67.74	97.43
3	0.17	0.30	0.00	3.00	5.84	10.32	10.82	24.08	38.83	51.27	71.08	102.40
4	3.85	5.17	3.26	0.00	0.89	8.07	9.73	26.85	44.59	58.76	81.80	119.02
5	7.41	9.27	6.84	0.96	0.00	5.98	8.03	24.99	43.25	57.51	82.05	121.66
6	10.76	12.57	11.15	6.86	5.09	0.00	1.26	8.38	21.75	34.55	59.52	98.91
7	10.82	12.48	11.40	8.31	6.89	1.18	0.00	6.05	17.97	28.84	51.47	88.01
8	21.67	22.69	23.04	23.49	22.66	8.66	6.25	0.00	5.66	15.32	37.14	71.77
9	39.45	40.27	41.55	44.32	43.77	24.31	20.58	6.15	0.00	4.64	21.34	53.38
10	62.55	63.84	65.20	67.10	65.41	42.94	37.72	19.55	5.79	0.00	11.73	39.80
11	105.60	106.88	108.30	112.67	110.92	84.67	77.65	54.38	30.20	15.26	0.00	16.75
12	183.01	182.65	185.72	195.56	195.18	163.80	151.83	120.40	88.26	61.62	25.25	0.00

(f) Layerwise KL divergence ($\times 10^{-2}$) on MNLI.

Figure 2: Layerwise KL Divergence of BERT$_{base}$ hidden states. The numbers on the top and the left of each sub-figure are the indices for the intermediate layers.

Distillation Objectives	SST-2	MRPC	QQP	MNLI-m/mm	QNLI	RTE	Avg
$\mathcal{L}_{hard} + \mathcal{L}_{soft}$	90.0	87.1	86.0	81.0/81.8	88.1	64.6	82.7
+ Reusing	90.4	87.7	86.6	81.4/81.8	88.4	65.0	83.0
$\mathcal{L}_{soft} + \mathcal{L}_{layer}$	89.9	88.1	86.3	81.7/81.5	87.8	66.4	83.1
+ Reusing	91.3	88.4	86.6	81.9/81.7	87.9	66.4	83.5
$\mathcal{L}_{hard} + \mathcal{L}_{layer}$	89.1	87.4	86.0	79.8/80.6	86.1	68.2	82.5
+ Reusing	89.2	87.3	86.6	80.2/80.5	86.7	63.9	82.1
$\mathcal{L}_{hard}$	89.2	85.9	86.0	80.4/80.1	86.8	64.6	81.9
+ Reusing	89.4	85.9	86.4	80.6/80.4	86.9	65.0	82.1
$\mathcal{L}_{soft}$	89.9	85.9	86.3	81.0/82.4	87.0	63.9	82.3
+ Reusing	90.3	86.5	86.9	81.4/82.0	88.0	63.2	82.6
$\mathcal{L}_{layer}$	50.9	81.2	55.8	36.5/36.7	39.8	47.3	49.7
+ Reusing	89.9	85.5	80.9	72.2/73.7	78.5	58.5	77.0

Table 8: The effectiveness of reusing the teacher classifier using different distillation objectives. *Reusing* in the table refers to the method of reusing the pre-trained teacher's classifier described in Section 3.2. Results are reported on 6-layer students.

Layer Mapping Strategy	SST-2	MRPC	QQP	MNLI-m/mm	QNLI	RTE	Avg
Top	90.1	87.0	86.2	81.5/82.2	87.6	63.9	82.6
Uniform	91.1	88.6	86.6	81.7/82.4	87.6	60.6	82.7

Table 9: Comparison between the top layer mapping strategy and the uniform layer mapping strategy. Results are reported on 6-layer students.