# ReAugKD: Retrieval-Augmented Knowledge Distillation For Pre-trained Language Models

**Jianyi Zhang[1], Aashiq Muhamed[2], Aditya Anantharaman[2], Guoyin Wang[2], Changyou Chen[3], Kai Zhong[2], Qingjun Cui[2], Yi Xu[2], Belinda Zeng [2], Trishul Chilimbi [2], Yiran Chen [1]**
{jianyi.zhang,yiran.chen}@duke.edu, cchangyou@gmail.com,
{muhaaash,aditanan,guoyiwan,kaizhong,qingjunc,zengb,trishulc}@amazon.com,
[1] Duke University, [2] Amazon [3] University at Buffalo, SUNY

## Abstract

Knowledge Distillation (KD) (Hinton et al., 2015) is one of the most effective approaches for deploying large-scale pre-trained language models in low-latency environments by transferring the knowledge contained in the large-scale models to smaller student models. Previous KD approaches use the soft labels and intermediate activations generated by the teacher to transfer knowledge to the student model parameters alone. In this paper, we show that having access to non-parametric memory in the form of a knowledge base with the teacher's soft labels and predictions can further enhance student capacity and improve generalization. To enable the student to retrieve from the knowledge base effectively, we propose a new Retrieval-augmented KD framework with a loss function that aligns the relational knowledge in teacher and student embedding spaces. We show through extensive experiments that our retrieval mechanism can achieve state-of-the-art performance for task-specific knowledge distillation on the GLUE benchmark (Wang et al., 2018a).

## 1 Introduction

Large pre-trained language models, such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and Electra (Clark et al., 2020) have achieved significant success on several different NLP tasks (Ding et al., 2019; Wang et al., 2018a) with fine-tuning. However, these models usually contain millions and billions of parameters, preventing their execution on resource-restricted devices. To deploy these models, Knowledge distillation (KD) is an effective compression technique to derive a smaller student model from a larger teacher model by transferring the knowledge embedded in the teacher's network. Previous KD methods typically store knowledge in the student's parameters and train the student by minimizing divergence between the student's and teacher's output prediction and intermediate activation distributions (Park et al., 2019; Zhang et al., 2018). However, the student's parametric memory is often limited and cannot be quickly expanded or revised. Moreover, after training, the teacher model's soft labels and activations, which contain essential task-specific knowledge, are not utilized by the student at inference time.

To address the issues mentioned above, we propose the *Retrieval-augmented Knowledge Distillation* (ReAugKD) framework. ReAugKD introduces a non-parametric external memory in addition to the implicit parametric memory of the model and uses kNN retrieval to retrieve from this memory. The key intuition of ReAugKD is to enhance the effective capacity of the student by using an external memory derived from relevant task-specific knowledge of the teacher. While this external memory could include any task-specific knowledge, in this work, it is composed of the soft labels and embeddings generated by the teacher model.

Our framework consists of an inference phase and a training phase. In the inference phase, we aggregate the soft labels of those teacher embeddings in our memory that are most similar to the student embedding. We demonstrate the efficacy of our framework by achieving state-of-the-art results on the GLUE benchmark (Wang et al., 2018a) with less than 3% latency overhead over the baseline without retrieval augmentation. ReAugKD also comprises a training phase, where we train the student to retrieve from the external memory effectively. We train with a novel relational KD loss that minimizes the divergence between teacher-teacher and teacher-student embedding distributions. We not only observe that training with this loss is necessary to align the student and teacher embedding spaces for retrieval but also that this loss improves student generalization even in the absence of retrieval augmentation. This suggests that incorporating the ability to retrieve information can significantly enhance generalization during the process of knowledge distillation.

In summary, our contributions include

- We propose ReAugKD, a novel framework for knowledge distillation that introduces a non-parametric memory to increase the effective student size. We show that retrieving from a memory composed of training set teacher predictions at inference time can significantly improve generalization on the GLUE tasks.
- To effectively retrieve from the non-parametric memory, we introduce a novel loss function that transfers the relational knowledge between teacher-teacher embedding and teacher-student embedding distribution. This loss function improves student generalization even in the absence of retrieval augmentation at inference time.
- We study the accuracy and latency cost with the number of neighbors (k) retrieved in ReAugKD. ReAugKD with approximate kNN introduces a small overhead of <3% latency increase.

## 2 Related Work

**Knowledge distillation** KD can be broadly classified into task-specific KD, where the student model will be used for the same task as the teacher model (Mirzadeh et al., 2020; Jin et al., 2019; Zhang et al., 2018; Sun et al., 2019) and task-agnostic KD where the student may be used for a different task, after finetuning on the new task (Jiao et al., 2019; Sun et al., 2020; Sanh et al., 2019; Wang et al., 2020; Zhang et al., 2018; Xu et al., 2019). In this work, we show that ReAugKD can be applied to enhance task-specific distillation as well as when finetuning task-agnostic distilled models. Closest to our work is RKD (Park et al., 2019) that introduces a loss to transfer relational knowledge between teacher-teacher embedding and student-student embedding distributions. Our work differs in that we transfer relational knowledge between teacher-teacher embedding and teacher-student embedding distribution to enhance the student model's ability to retrieve from the external memory. MetaDistil (Zhou et al., 2022) is a strong task-specific distillation baseline that employs meta-learning to better transfer knowledge to the student. Unlike MetaDistill, we show that ReAugKD can significantly improve the student model's generalization without retraining the whole teacher with meta-learning.

**Retrieval-augmented language models** There has been growing interest in retrieval-augmented methods for Knowledge-Intensive generative NLP Tasks, such as text generation and question answering (Weston et al., 2018; Lewis et al., 2020; Guu et al., 2020; Lin et al., 2022), where querying training examples during inference significantly improves likelihood. Closest to our work is BERT-kNN (Kassner and Schütze, 2020) which combines BERT with a kNN search over a large datastore of an embedded text collection, to improve cloze-style QA. In our work, we apply retrieval augmentation to enhance the capacity of student models during KD, and show improvement even on non-knowledge intensive tasks like GLUE.

## 3 Methodology

### 3.1 Training Phase

Our framework consists of two main phases, the training phase and the inference phase. The training phase has two steps. In the first step, we prepare the teacher model for KD by adding a linear projection head $\mathcal{L}$ on the top of the teacher model encoder that has been finetuned for a specific downstream task. The input dimension of this projection head is the embedding dimension of the teacher. The output dimension is the embedding dimension of the student. We then freeze the other parameters of the teacher model and finetune the parameters in $\mathcal{L}$ with supervised contrastive loss (Khosla et al., 2020). This step a) reduces the dimension of the teacher's embeddings, to the student model dimension for retrieval, and b) uses supervised contrastive loss to derive a kNN classifier for BERT that is robust to natural corruptions, and hyperparameter settings (Li et al., 2021). Fine-tuning $\mathcal{L}$ also greatly reduces the computational cost compared to retraining the whole teacher model (Zhou et al., 2022).

In the second step, we perform KD by generating the teacher embeddings with $\mathcal{L}$ and teacher soft labels using the original teacher's classifier head for a batch of data. Then, we use the loss function we proposed in Section 3 to train our student model.

### 3.2 Loss function

We present some mathematical notations to introduce our loss function. Given a batch of data $\{d_i\}, i = 1, 2, \cdots, N$, where $N$ is the batch size, we denote the embedding generated by the teacher's projection head as $z_i$ and the soft labels generated by the teacher's classifier as $\bar{y}_i$. Similarly, we adopt $x_i, y_i$ to denote the student's embeddings and predictions. Then we construct a probability distribution $q_{i,j}$ over each teacher's embeddings $z_j$

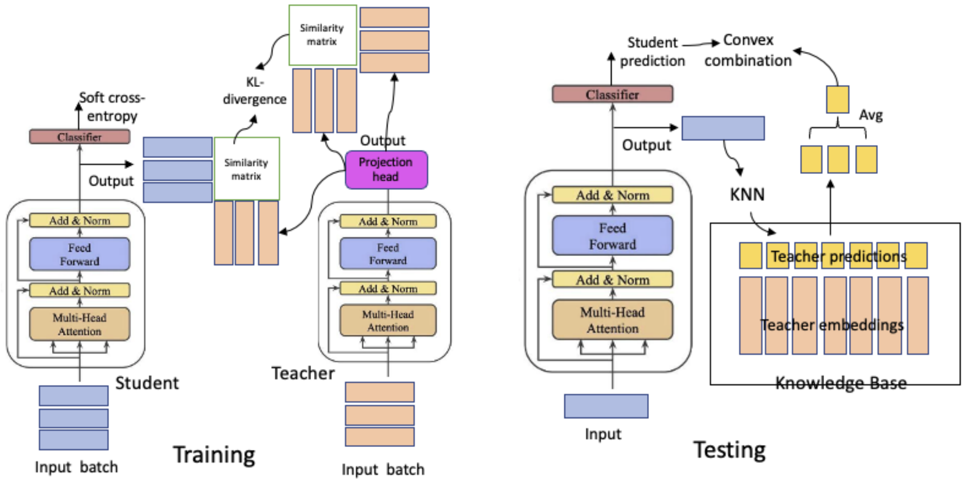


Figure 1: Training and Inference (Testing) phases of Retrieval-augmented Knowledge Distillation (ReAugKD).

to capture the similarity with respect to an anchor point $z_i$,

$$q_{i,j} = \frac{\exp{(z_i \cdot z_j)}/\tau}{\sum_{k=1}^{N} \exp{(z_i \cdot z_k)}/\tau}, \quad (1)$$

where the $\tau$ stands for temperature. Note that $\sum_{j=1}^{N} q_{i,j} = 1$. $q_{i,j}$ reflects the cosine distance relational knowledge among different embeddings generated by the teacher model in the batch. If $z_j$ is closer to $z_i$, cosine distance, $q_{i,j}$ will be larger. Similarly, given a student's embedding $x_i$ as an anchor point, we formulate another probability distribution $\bar{q}_{i,j}$ over each teacher's embeddings $z_j$ of the data in the batch.

$$\bar{q}_{i,j} = \frac{\exp{(x_i \cdot z_j)}/\tau}{\sum_{k=1}^{N} \exp{(x_i \cdot z_k)}/\tau}. \quad (2)$$

The $\bar{q}_{i,j}$ reflects the cosine distance relationship between different embeddings generated by the teacher model and the student's embedding. Our loss function aims to minimize the divergence of these two distributions $\bar{q}_{i,j}$ and $q_{i,j}$ since the teacher model is a strong kNN classifier after finetuning with supervised contrastive loss function in the first step of our training. In the ideal case, given a student's embedding $x_i$, the student retriever should retrieve the same set of embeddings as the corresponding teacher's embedding $z_i$. We adopt KL divergence to measure that divergence. In addition, we adopt the commonly-used cross-entropy loss to calculate the divergence between the student's prediction $y_i$ and the teacher's prediction $\bar{y}_i$.

Our loss function can be formulated as

$$CE(y_i, \bar{y}_i) + \alpha KL(q_{i,j}, \bar{q}_{i,j}), \quad (3)$$

where $CE$ is the cross entropy loss and $KL$ is KL-divergence. $\alpha$ is the hyperparameter controlling the trade-off between the two losses.

### 3.3 Inference Phase

After training, we construct a knowledge base (KB) comprising of projected teacher embeddings and predictions. Given new data $d_i$ at inference time, we obtain $(x_i, y_i)$ using the student model. and use the HNSW algorithm (Malkov and Yashunin, 2018) to derive the $K$ nearest teacher's embeddings and their corresponding soft labels $\{(z_k, \bar{y}_k)\}_{i=1,2,\cdots,K}$ from the KB. Then we compute the weighted average of these soft labels $Avg(\{\bar{y}\})_i$ based on $\bar{q}_{i,k}$

$$Avg(\{y\})_i = \sum_{k=1}^{K} \frac{\bar{q}_{i,k}}{\sum_{k=1}^{K} \bar{q}_{i,k}} \bar{y}_k$$

We derive a new prediction $\bar{y}'_i$ for $d_i$ with $Avg(\{\bar{y}\})_i$.

$$\bar{y}'_i = \beta \bar{y}_i + (1 - \beta) Avg(\{\bar{y}\})_i,$$

$\beta$ is the hyperparameter controlling the trade-off between the two predictions.

## 4 Experimental Results

We apply our method to distill BERT-Base (Devlin et al., 2018) into a 6-layer BERT with a hidden size of 768. We evaluate our proposed approach, ReAugKD, on the GLUE benchmark (Wang et al., 2018a). These datasets can be broadly divided into three families of problems: single-set tasks that include linguistic acceptability (CoLA) and sentiment analysis (SST-2), similarity, and paraphrasing tasks (MRPC and QQP); inference tasks

| Method | #Param | GLUE | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | CoLA (8.5k) | QNLI (105k) | QQP (364k) | RTE (2.5k) | SST-2 (67k) | MRPC (3.7k) | Avg |
| BERT-Base (teacher) (Devlin et al., 2018) | 110M | 58.9 | 91.2 | 91.4 | 71.4 | 93.0 | 87.6 | 82.25 |
| BERT-6L (student)(Turc et al., 2019) | 66M | 53.5 | 88.6 | 90.4 | 67.9 | 91.1 | 84.4 | 79.32 |
| Task-specific Distillation | | | | | | | | |
| KD (Hinton et al., 2015) | 66M | 54.1 | 89.2 | 90.9 | 67.7 | 91.2 | 85.2 | 79.72 |
| PKD (Sun et al., 2019) | 66M | 54.5 | 89.5 | 90.9 | 67.6 | 91.3 | 84.7 | 79.75 |
| TinyBERT w/o DA (Jiao et al., 2019) | 66M | 52.4 | 89.8 | 90.6 | 67.7 | 91.9 | 86.5 | 79.82 |
| RCO (Jin et al., 2019) | 66M | 53.6 | 89.7 | 90.6 | 67.6 | 91.4 | 85.1 | 79.67 |
| TAKD (Mirzadeh et al., 2020) | 66M | 53.8 | 89.6 | 90.7 | 68.5 | 91.4 | 85.0 | 79.83 |
| RKD (Park et al., 2019) | 66M | 53.4 | 89.5 | 90.9 | 68.6 | 91.7 | 86.1 | 80.03 |
| DML (Zhang et al., 2018) | 66M | 53.7 | 89.6 | 90.3 | 68.4 | 91.5 | 85.1 | 79.77 |
| ProKT (Shi et al., 2020) | 66M | 54.3 | 89.7 | 90.9 | 68.4 | 91.3 | 86.3 | 80.15 |
| SFTN (Park et al., 2021) | 66M | 53.6 | 89.5 | 90.4 | 68.5 | 91.5 | 85.3 | 79.80 |
| MetaDistil (Zhou et al., 2022) | 66M | 58.6 | 90.4 | 91.0 | 69.4 | 92.3 | **86.8** | 81.42 |
| ReAugKD (ours) | 66M | **59.4** | **90.7** | **91.24** | **70.39** | **92.5** | 86.3 | **81.76** |
| ReAugKD w/o retrieval | 66M | 59.1 | 90.6 | 91.21 | 69.31 | 92.3 | 85.8 | 81.39 |

Table 1: Experimental results of ReAugKD and other previous works on the development set of GLUE. Numbers under each dataset indicate the number of training samples. The results of the baselines are from (Zhou et al., 2022). We report Matthew's correlation coefficient for CoLA and accuracy for other datasets.

that include Natural Language Inference (MNLI and RTE); and Question Answering (QNLI). We compare our method with vanilla KD (Hinton et al., 2015), TAKD (Mirzadeh et al., 2020), RCO (Jin et al., 2019), RKD (Park et al., 2019), DML (Zhang et al., 2018), PKD (Sun et al., 2019) ProKT (Shi et al., 2020), SFTN (Park et al., 2021) and MetaDistil (Zhou et al., 2022). Following similar setting as MetaDistill, we perform a grid search over the sets of the weight of KD loss from {0.9, 0.99}, the predictions weight $\beta$ from {0, 0.1, ... 1} and the top-$k$ from 1 to 20. We set the student learning rate to 2e-5 and the batch size to 64.

**Experimental Results on GLUE** We report the experimental results on the development set of the six GLUE tasks in Table 1. Notably, our method achieves start-of-the-art results on five out of the six datasets with an average improvement of 0.34% over the previous best KD method MetaDistil (Zhou et al., 2022). Although MetaDistil achieves slightly better performance on the MRPC dataset, our method has the advantage of not needing to conduct meta-learning on the whole large teacher model, which significantly increases extra training cost in terms of time and memory (Zhou et al., 2022). In addition, we also observe a performance gain of 0.37% with the retrieval component of ReAugKD as compared to ReAugKD without retrieval which verifies the benefit of retrieval augmentation in our approach. Even without the retrieval process, the student model trained by our

| Method | QNLI | | SST-2 | | CoLA | |
|---|---|---|---|---|---|---|
| | acc | time | acc | time | mcc | time |
| ReAugKD w/o Retrieval | 90.6 | 45.70s | 92.3 | 7.80s | 59.1 | 8.67s |
| ReAugKD (k=5) | 90.72 | +1.31s | 92.43 | +0.199s | 58.87 | +0.143s |
| ReAugKD (k=10) | 90.70 | +1.32s | 92.54 | +0.201s | 59.39 | +0.147s |
| ReAugKD (k=15) | 90.74 | +1.33s | 92.54 | +0.202s | 59.35 | +0.147s |
| ReAugKD (k=20) | 90.72 | +1.33s | 92.43 | +0.204s | 59.37 | +0.148s |

Table 2: Analysis of the sensitivity of top k on model performance and retrieval time

designed loss can still achieve comparable performance to MetaDistill on most datasets. Since our loss is designed to improve the student retrieval function, this demonstrates the importance of retrieval capability in KD.

**Number of Neighbors Retrieved (k)** To understand the time overhead of retrieval on the student model's inference time, we investigate the performance and additional time overhead of the retrieval process while varying the number of neighbors retrieved (k) in Table 2. From the results, it is clear that retrieval improves the student model performance with an additional time overhead of less than 3% of the original inference time. The retrieval process is conducted only on CPU, and does not take up GPU resources during training.

## 5 Conclusion

In this paper, we present ReAugKD, a knowledge distillation framework with a retrieval mechanism that shows state-of-the-art performance on the GLUE benchmark. In the future, we plan to expand the knowledge base with more information from the teacher and extend it to additional tasks.

**Limitations** Our method relies on having access to teacher embeddings and prediction which may not always be possible in a black-box distillation setting. Retrieval augmentation also requires maintaining a knowledge base that is memory intensive. The cost of the retrieval process is dependent on the size of the training corpus, which can be a limitation when dealing with very large training datasets. Conducting dataset distillation (Wang et al., 2018b) on the training corpus to further reduce memory cost and retrieval time is an important future step for our framework.

**Acknowledgments** This work was done when Jianyi Zhang was an intern at Amazon Search. In addition, Jianyi Zhang and Yiran Chen disclose support from grants CNS-2112562, IIS-2140247, and CNS-1822085. We thank Yuchen Bian for the valuable discussion and thank all reviewers for their valuable comments.

## References

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. *arXiv preprint arXiv:1905.05460*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. 2019. Knowledge distillation via route constrained optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1345–1354.

Nora Kassner and Hinrich Schütze. 2020. Bert-knn: Adding a knn search component to pretrained language models for better qa. *arXiv preprint arXiv:2005.00766*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Linyang Li, Demin Song, Ruotian Ma, Xipeng Qiu, and Xuanjing Huang. 2021. Knn-bert: fine-tuning pre-trained models with knn classifier. *arXiv preprint arXiv:2110.02523*.

Bill Yuchen Lin, Kangmin Tan, Chris Miller, Beiwen Tian, and Xiang Ren. 2022. Unsupervised cross-task generalization via retrieval augmentation. *arXiv preprint arXiv:2204.07937*.

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*.

Yu A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*, 42(4):824–836.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198.

Dae Young Park, Moon-Hyun Cha, Daesin Kim, Bohyung Han, et al. 2021. Learning student-friendly teacher networks for knowledge distillation. *Advances in Neural Information Processing Systems*, 34:13292–13303.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Wenxian Shi, Yuxuan Song, Hao Zhou, Bohan Li, and Lei Li. 2020. Learning from deep model via exploring local targets.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.

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

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018a. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. 2018b. Dataset distillation. *arXiv preprint arXiv:1811.10959*.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2020. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. *arXiv preprint arXiv:2012.15828*.

Jason Weston, Emily Dinan, and Alexander H Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*.

Yuhui Xu, Yuxi Li, Shuai Zhang, Wei Wen, Botao Wang, Wenrui Dai, Yingyong Qi, Yiran Chen, Weiyao Lin, and Hongkai Xiong. 2019. Trained rank pruning for efficient deep neural networks. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC2-NIPS)*, pages 14–17.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328.

Wangchunshu Zhou, Canwen Xu, and Julian McAuley. 2022. Bert learns to teach: Knowledge distillation with meta learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7037–7049.

# A Appendix

## A.1 ReAugKD with task-agnostic distillation

| Model | #Param | QNLI | QQP | RTE | SST-2 | MRPC | MNLI-m | CoLA | Avg |
|---|---|---|---|---|---|---|---|---|---|
| Teacher Model (24 $\times$ 1024 RoBERTa-large (Liu et al., 2019)) | | | | | | | | | |
| RoBERTa-large | 354M | 94.7 | 92.2 | 86.6 | 96.4 | 90.9 | 90.2 | 68 | 88.43 |
| Distilled Student Model (6×768 MiniLMv2) | | | | | | | | | |
| Pretraining Distillation | 81M | 92.7 | 91.4 | 78.7 | 94.5 | **90.4** | 87.0 | 54.0 | 83.8 |
| ReAugKD | 81M | **93.1** | **91.9** | **80.5** | **95.0** | 90.2 | **88.5** | **57.9** | **85.30** |
| ReAugKD w/o Retrieval | 81M | 93.0 | 91.8 | 79.8 | 94.9 | 90.2 | 88.3 | 57.2 | 85.02 |

Table 3: Results of our method improving finetuned task performance of MiniLMv2

Previous results have demonstrated the effectiveness of our method for task-specific distillation. Our method can further improve the finetuned performance of task-agnostic distilled models. We adopt RoBERTa-large as the teacher model and the MiniLMv2 as the student model to verify the effectiveness of our method. Our method can achieve around $2\%$ improvement in performance.

## A.2 Details about training teacher model's projection head

We adopt the $L_{out}^{sup}$ version of the loss function in (Khosla et al., 2020) to finetune the parameters of the projection head, which is

$$L_{out}^{sup} = -\sum_{i=1}^{N} \frac{1}{N} \sum_{j \in P(i)} log \frac{\exp\left(z_i \cdot z_j\right)/\tau}{\sum_{k=1}^{N} \exp\left(z_i \cdot z_k\right)/\tau}. \tag{4}$$

Here, there are $N$ data samples $d_i$ in the batch and we denote the embedding generated by the teacher's projection head for the $i$-th data $d_i$ as $z_i$. $P(i)$ here represents the set of all the positive data samples for data $d_i$. The data samples from the same class are considered as positive pairs and the data samples from different classes are considered as negative pairs. Regarding the use of data augmentation in training the projection head, we chose not to adopt data augmentation as we found that using supervised contrastive loss without data augmentation was sufficient to achieve results comparable to the cross-entropy loss used in supervised learning. We use the AdamW optimizer with a learning rate of 0.00002. The batch size was set to 512, and the temperature for the supervised contrastive loss (SCL) was set to 0.07. We trained the model 3 epochs.