Pro-KD: Progressive Distillation by Following the Footsteps of the Teacher

Mehdi Rezagholizadeh[1*]   **Aref Jafari**[2,1*]   **Puneeth S.M. Saladi**[1 †]
Pranav Sharma[2]   **Ali Saheb Pasand**[1,2]   **Ali Ghodsi**[2]
[1]Huawei Noah's Ark Lab
[2]University of Waterloo

{mehdi.rezagholizadeh, puneeth.saladi}@huawei.com

{aref.jafari, p68sharm, asahebpa, ali.ghodsi}@uwaterloo.ca

Abstract

With the ever growing scale of neural models, knowledge distillation (KD) attracts more attention as a prominent tool for neural model compression. However, there are counter intuitive observations in the literature showing some challenging limitations of KD. A case in point is that the best performing checkpoint of the teacher might not necessarily be the best teacher for training the student in KD. Therefore, one important question would be how to find the best checkpoint of the teacher for distillation? Searching through the checkpoints of the teacher would be a very tedious and computationally expensive process, which we refer to as the *checkpoint-search problem*. Moreover, another observation is that larger teachers might not necessarily be better teachers in KD, which is referred to as the *capacity-gap* problem. To address these challenging problems, in this work, we introduce our progressive knowledge distillation (Pro-KD) technique which defines a smoother training path for the student by following the training footprints of the teacher instead of solely relying on distilling from a single mature fully-trained teacher. We demonstrate that our technique is quite effective in mitigating the capacity-gap problem and the checkpoint search problem. We evaluate our technique using a comprehensive set of experiments on different tasks such as image classification (CIFAR-10 and CIFAR-100), natural language understanding tasks of the GLUE benchmark, and question answering (SQuAD 1.1 and 2.0) using BERT-based models and consistently got superior results over state-of-the-art techniques.

1 Introduction

Knowledge distillation (KD) (Hinton et al., 2015) has gained a lot of attention in different deep learn-

Table 1: The best performing checkpoints vary for different size of the models, and different tasks. It is evident that the best checkpoint of the teacher does not necessarily lead to the best performing student model.

Task	Model	Best #ChPt Epoch	ACC	Δ ↑
MRPC	BERT$_{LARGE}$(♠T)	6	88.22	-
	BERT$_{SMALL}$(♣S)	3	85.29	+0.7
	DistilBERT(♣S)	5	89.16	+0.8
SST-2	BERT$_{LARGE}$(♠T)	1	92.89	-
	BERT$_{SMALL}$(♣S)	7	88.76	+0.6
	DistilBERT(♣S)	7	91.63	+0.4
QNLI	BERT$_{LARGE}$(♠T)	6	92.4	-
	BERT$_{SMALL}$(♣S)	2	87.08	+0.3
	DistilBERT(♣S)	4	90.46	+0.2

♠T:Teacher, ♣S: Student, #ChPt: Checkpoint Number, ACC: Accuracy, Δ ↑: Accuracy improvement compared to the best checkpoint of the teacher

ing applications such as natural language processing (NLP) (Sun et al., 2019; Jiao et al., 2019; Clark et al., 2019), computer vision (Guo et al., 2020; Mirzadeh et al., 2019), and speech processing (Yang et al., 2019; Yoon et al., 2020; Chebotar and Waters, 2016). Nowadays, the scale of neural networks is growing in the favor of improving their performance (Devlin et al., 2019). A case in point is pre-trained language models (PLMs) such as the GPT-3 (Brown et al., 2020), Pangu-α (Zeng et al., 2021), and WuDao2 which have more than a hundred billions of parameters (Brown et al., 2020). However, deploying these models on devices with limited computational power will be very challenging, if not impossible. In this regard, KD can be used as one of the most prominent neural model compression techniques.

KD adds a new loss term to the regular cross-entropy classification loss. This new loss encourages the student model to mimic the output of a pre-trained teacher network. The teacher network is usually a higher capacity model which is able to learn the underlying function of the training data to

---

4714

a good extent. The output prediction of the teacher is called *soft-target* for the student model. In contrast to ground-truth labels coming from the training dataset which only carry the information about a single class, soft-targets can provide some information about the relative distribution of different classes for each training data. Therefore, a pretrained teacher is able to provide some auxiliary signal besides the labels in the training dataset.

KD has been investigated a lot in the literature. Sun et al. (2019); Passban et al. (2021); Wu et al. (2020) proposed a technique to improve KD by incorporating the intermediate layer matching in the KD loss. Jiao et al. (2019) show a two-stage KD with intermediate layer mapping, attention distillation and embedding distillation for BERT-based models. Mate-KD (Rashid et al., 2021) and MiniMax-KNN KD (Kamalloo et al., 2021) tailor data augmentation for KD, in which augmented samples are generated or selected based on maximum divergence loss between the student and teacher networks. Rashid et al. (2020) propose a zero-shot KD technique in NLP in which the student does not need to access the teacher training data for its training. Clark et al. (2019) use KD for multi-task learning in natural language understanding. Kim and Rush (2016) propose a sequence-level KD solution for machine translation. Guo et al. (2020) introduces a collaborative training of students with different capacities with KD.

Although KD has been successful in many different deep learning tasks, it is subject to some special limitations as well. For example, it is shown in (Lopez-Paz et al., 2015) that, based on VC theory (Vapnik, 1998), the teacher capacity should not be too large in KD. Similar observation is given by Mirzadeh et al. (2019) using empirical and theoretical justifications that KD will be less effective when the capacity gap between the teacher and student is large. This problem is referred to as the *capacity-gap* problem. Mirzadeh et al. (2019) proposed the TAKD solution to this problem by adding one or multiple intermediate teacher assistant (TA) networks to learn from the teacher using KD and then train the student using other distillation processes. However, training intermediate networks can be prohibitive in terms of adding to the training time, computational complexity and extra error propagation. The other case in point is that a fully-trained teacher might not be the best teacher for the student (Cho and Hariharan, 2019). In other words, an early-stopped teacher can be a better option for KD compared to a fully-trained one. This observation implies that we need to search through the checkpoints of the teacher to find the best model for the distillation process. However, this search can be very expensive especially when we deal with PLMs. We refer to this problem as the *checkpoint-search* problem. Our investigations of this checkpoint search problem show the significance of this issue especially dealing with PLMs. Table 1 depicts that the best distillation checkpoint of the teacher varies for each task and for each student configuration and it is very different from the teacher best performing checkpoint.

In this work, we propose our Pro-KD solution to tackle both the capacity-gap and checkpoint-search problems. In Pro-KD, the student grows gradually with the growth of the teacher. We hypothesize that the training path of the teacher can be informative for the student and we should not disregard it. Therefore, in contrast to the original KD where the student learns from the best pre-trained teacher, in Pro-KD the student starts its learning process together with the teacher. Furthermore, in contrast to the TAKD technique which reduces the capacity of the teacher by adding intermediate TAs to the distillation process, in our Pro-KD, we mitigate the capacity gap by making the training path of the student more smooth and gradual by following the training footsteps of the teacher. Moreover, to make the training smooth further for the student, inspired by (Jafari et al., 2021), we apply an adaptive temperature factor to the output of the teacher while being trained. This temperature factor is decreased during the training. We will show the effectiveness of our solution using theoretical justification and empirical evaluations. We evaluate Pro-KD by performing experiments on both NLP (the GLUE (Wang et al., 2018b) benchmark and SQuAD 1.1 and 2.0) and image classification tasks (CIFAR-10, CIFAR-100). The contributions of this paper are summarized in the following:

1. We propose our Pro-KD solution to the KD *capacity-gap* and *checkpoin-search* problems. In Pro-KD, the student follows the training footsteps of the teacher. Moreover, it introduces a dynamic temperature function to the output of the teacher when it is distilled to the student to make the student training gradual and smooth.

2. We apply our technique to ResNET8 model on

both CIFAR-10 and CIFAR-100 image classification tasks, and the natural language inference task on different BERT based models such as DistilRoBERTa, and BERT-Small on the GLUE benchmark and also SQuAQ 1.1 and 2.0 question answering datasets and achieved the state-of-the-art results.

3. Our technique is simple, architecture agnostic, does not require training any extra network and can be applied on top of different variants of KD.

2 Background

Knowledge Distillation KD (Hinton et al., 2015) is a well-known method for neural model compression and also is shown to be an effective regularizer in improving the performance of neural networks in the self-distillation (Yun et al., 2020; Hahn and Choi, 2019) or born-again (Furlanello et al., 2018) setups. KD adds a particular loss term to the regular cross entropy (CE) classification loss:

$$\mathcal{L}_{KD}(\phi) = CE\Big(y, S(x;\phi)\Big) +$$
$$\mathcal{T}^2 KL\Big(\sigma(\frac{z_t(x;\theta)}{\mathcal{T}}), \sigma(\frac{z_s(x;\phi)}{\mathcal{T}})\Big) \quad (1)$$

where x is the input data and y is its associated label, ϕ and θ refer to the student and teacher parameters, σ is the softmax function, z_s and z_t are the student and teacher logits, $\mathcal{T}$ is the temperature parameter to control the softness of the output probability distributions, CE and KL refer to the cross entropy and KL divergence loss functions respectively.

Regular KD training is a two-stage process in which the teacher is fully trained in the first stage and deployed in training of the student model in the next stage. The student is trained based on the *hard labels* coming from the ground-truth training data and *soft labels* coming from the teacher output predictions.

3 Related Work

In this section, we review the most related works to our paper in the literature.

3.1 Capacity-Gap Problem in KD

The capacity-gap problem in KD, refers to having a more powerful or larger teacher is not necessarily leads to a better training for the student. Mirzadeh

et al. (2019) propose their TAKD solution to this problem by introducing an intermediate TA network whose capacity is greater than the student but smaller than the teacher. The target of this TA network is to learn from the teacher and train the student. It is evident that in this setup the capacity gap between the TA and student is less than that of the main teacher and the student. Even though adding the TA network can mitigate the capacity-gap problem, there are two downsides in this technique: first, training a separate TA network is costly; second, the sequence of training multiple networks can lead to error accumulation and error propagation to the student. Moreover, TAKD only showed their results on image classification tasks. Jafari et al. (2021) proposed a TA-free solution for the capacity-gap problem which is called Annealing-KD. Annealing-KD adds an annealed dynamic temperature factor to the output of the teacher to make the training process for the student very gradual. The temperature factor starts from a large value to apply the maximum smoothing to the output of the teacher at the beginning of the training and the gradually tends toward 1 where there is no smoothing effect on the output of the teacher. Our Pro-KD is inspired by Annealing-KD in using a dynamic temperature factor to solve the capacity-gap problem, but we can highlight two main improvements over Annealing-KD: first, Annealing-KD suffers from the checkpoint search problem while Pro-KD does not; second, Pro-KD takes advantage of the intrinsic gradual training of the teacher training which is not considered in Annealing-KD.

3.2 Checkpoint Search Problem in KD

Choi et al. (2020) highlighted their findings over computer vision (CV) models that bigger models are not necessarily better teachers and also early stopped teachers can train students better. To overcome these issues they propose an ad-hoc approach called Early Stopped KD (ESKD); however, ESKD still needs to search among the pool of teacher's early checkpoints and the selection mechanism among this early checkpoints is not clear. Moreover, in PLMs, generally teachers are not fine-tuned for long and yet searching among the checkpoints of large PLMs is expensive. (Jin et al., 2019) proposed Route Constrained Optimization (RCO) distillation which is a curriculum learning technique to follow easy-to-hard training scheme on top of

the teacher's training trajectory. We can deem this technique as the most related work to us with the following differences: 1- our Pro-KD has a temperature factor to make the training process more gradual which can handle the capacity gap better; 2- RCO still requires search to define its trajectory; 3- we evaluated Pro-KD on both CV and NLP tasks but RCO is only evaluated on CV tasks. We will show in the Experiment section that our Pro-KD outperforms RCO consistently.

In the next section, we introduce our Pro-KD solution to the capacity-gap and checkpoint-search problems. Pro-KD exploits a smooth training process for the student to gradually learn from the teacher while being trained from scratch, instead of learning from a fully-trained Teacher.

4 Methodology: Pro-KD

Methodology of this paper concerns addressing the capacity-gap and checkpoint-search problems in KD. We propose our Pro-KD technique which is a progressive training procedure in which the student is trained together with the teacher. Therefore in Pro-KD, the student is not exposed to the output of a fully trained teacher from beginning, and instead, it learns from the teacher at the same time while the the teacher is being trained. This training process for the student is more gradual and also the student can learn from the training path of the teacher. Inspired by Jafari et al. (2021), We define the training in two phases: first, the student is only supervised by an adaptive smoothed version of the teacher; second, the student will be only trained on ground-truth labels. The detail of each phase is explained in the following.

Phase I) General Step-by-Step Training with a Teacher In this phase, the teacher is being trained using the cross entropy loss on the training data. Without loss of generality, let's assume that $\theta(i)$, the optimized parameters of the teacher at the beginning of epoch i, will be updated during the ith epoch of training to obtain $\theta(i+1)$:

$$\mathcal{L}_T(\theta(i)) = CE\big(y, T(x; \theta(i))\big)$$
$$\theta(i+1) \leftarrow \min_{\theta} \mathcal{L}_T(\theta(i)) \tag{2}$$

Then, a smoothed version of the teacher output using a temperature factor $\mathcal{T}_i$ at epoch i will be used to train the student at epoch j:

$$\mathcal{L}_S^I(\phi(j)) = \|z_s(x; \phi(j)) - \frac{z_t(x; \theta(i))}{\mathcal{T}_i}\|_2^2$$
$$\phi(j+1) \leftarrow \min_{\phi} \mathcal{L}_S^I(\phi(j)) \tag{3}$$
$$\mathcal{T}_i \in \{\mathcal{T}_{i+1} = \mathcal{T}_i - 1 | \mathcal{T}_1 = \tau_{max}, \mathcal{T}_{i+1} \geq 1\}$$

where $\mathcal{L}_S^I$ refers to the student loss function in phase I, z_s and z_t are the logits of the student and teacher respectively, $\phi(j)$ represents the parameters of the student model at epoch j, $\mathcal{T}_i$ is adaptive temperature factor. The rational behind applying the temperature to the teacher output is to make the output of the teacher smoother for the student to learn, especially at early stages of the training process (Jafari et al., 2021).

It is evident that early-stopped teachers can be better teachers for KD (Cho and Hariharan, 2019). In that case, for an early-stopped teacher, the number of training epochs of the student can go longer than that of the teacher. Therefore, for each given epoch i of the teacher, the student can be trained for $n_i \geq 1$ epochs, subject to $\sum_{i=1}^{\tau_{max}} n_i = N$ where N is the preset total number of student training epochs. n_i is set to a constant integer number in most of our experiments but can be customized as well.

Bear in mind that we apply an adaptive temperature to the teacher logit in the student distillation loss, and this temperature starts from the highest τ_{max} value and decrease linearly with the epoch number of the teacher throughout training. τ_{max} is a hyper-parameter in our training. Moreover, we keep the temperature fixed at each epoch i. After training with the teacher logits, in the next phase the student model will be trained on the ground-truth labels.

Phase II) Training with the Ground-Truth Labels After training the student based on the soft targets of the teacher, the student is trained on the ground-truth labels using the cross entropy loss for a few epochs:

$$\mathcal{L}_S^{II}(\phi(j)) = CE\big(y, S(x; \phi(j))\big)$$
$$\phi(j+1) \leftarrow \min_{\phi} \mathcal{L}_S^{II}(\phi(j)). \tag{4}$$

In summary, our technique enables the student to learn from the teacher more smoothly. Pro-KD is different from regular KD technique in the following aspects:
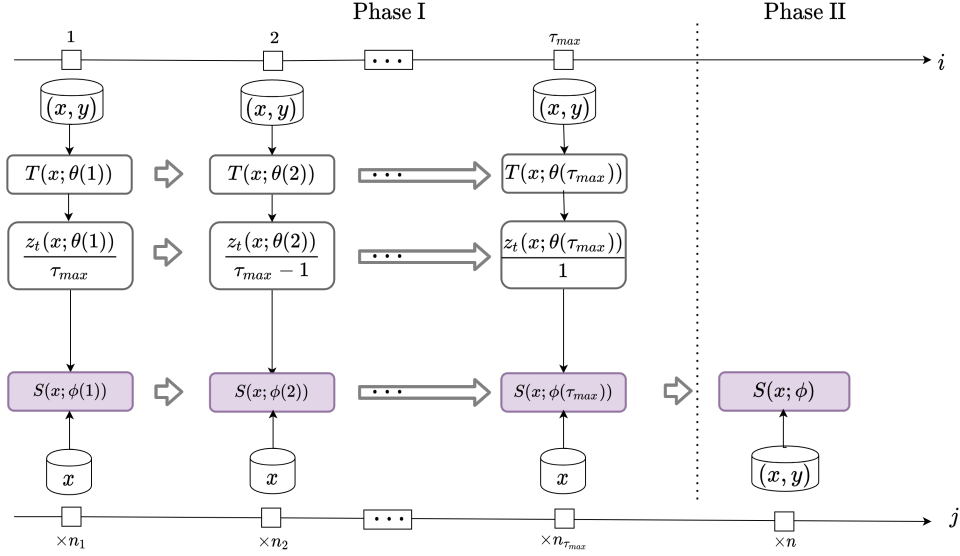
Figure 1: Phase I and Phase II of the Pro-KD method. Phase I: the teacher is trained on the labeled training data. The student at each step tries to mimic the behavior of the corresponding teacher checkpoint. The logits of the teacher at each time step is attenuated with the temperature parameter. We start training of the student from $\mathcal{T} = \tau_{max}$ and go to $\mathcal{T} = 1$. Phase II: training the student with the labeled data only using the cross entropy loss. n_i for $1 \leq i \leq \tau_{max}$ refers to the number of training epochs of the student corresponding to the ith training epoch of the teacher. n_is should add up to N which is the preset total number of training epochs of the student. Bear in mind that since the teacher is usually early stopped, teacher is trained less longer than the student model.

1. rather than distilling from a fully trained teacher; we distill from the teacher during its training;

2. in contrast to regular KD which applies a fixed temperature parameter to both networks, we apply an adaptive temperature factor only to the logits of the teacher;

3. we apply the KD loss and the cross entropy loss in two separate phases.

Moreover, it is worth mentioning that the overall number of training steps of our model follows regular KD techniques. For example, if in regular KD baselines, the model is trained for 30 number of epochs, we train the total training time of phase I and phase II of our model to 30 epochs as well.

4.1 Why Does Pro-KD work?

Lopez-Paz et al. (2015) do an analysis based on VC-dimension theory to discuss the conditions under which KD works better than no-KD scenarios. It is shown that KD works if the following inequality holds:

$$O(\frac{|\mathcal{F}_s|_c + |\mathcal{F}_t|_c}{n^\alpha}) + \varepsilon_t + \varepsilon_l \leq O(\frac{|\mathcal{F}_s|_c}{\sqrt{n}}) + \varepsilon_s \quad (5)$$

where $\mathcal{F}_s$ and $\mathcal{F}_t$ are the function classes corresponding to the teacher and student; $|.|_c$ is a function class capacity measure; $O(.)$ is the estimation error of training the learner; ε_s is the approximation error of the best estimator function belonging to the $\mathcal{F}_s$ class with respect to the underlying function; ε_t is a similar approximation error for the teacher with respect to the underlying function; ε_l is the approximation error of the best student function with respect to the teacher function; n is the number of training samples, and $\frac{1}{2} \leq \alpha \leq 1$ is a parameter related to the difficulty of the problem. Smaller values of α indicate slower training and larger values correspond to faster learning rates of the student. Given Eq. 5 we observe that for larger teacher models, the value of the left hand side of the inequality might get higher than that of the right hand side, which implies that KD might not work as expected for large capacity teachers.

We analyzed the capacity-gap problem based on the inequality 5. To satisfy this inequality when the capacity of the teacher is large (or equivalently the capacity gap between the two networks is large), we can think about reducing the capacity of the teacher (eg. the TAKD technique (Mirzadeh et al., 2019) introduces a smaller TA network instead of a large teacher), or increase α. Increasing α is

Table 2: Comparing the test accuracy of Pro-KD, TAKD (Mirzadeh et al., 2019), Annealing-KD (Jafarpour et al., 2021), RCO (Jin et al., 2019), regular KD, and student without teacher on CIFAR-10 dataset with both ResNet and CNN models

Model	Type	Training method	Accuracy
	Teacher(110)	from scratch	93.8
	TA(20)	KD	92.39
ResNet	Student(8)	from scratch	88.44
	Student(8)	KD	88.45
	Student(8)	TAKD	88.47
	student(8)	RCO	88.90
	Student(8)	Annealing-KD	89.44
	Student(8)	**Pro-KD (ours)**	**90.01**

Table 3: Comparing the test accuracy of Pro-KD, TAKD (Mirzadeh et al., 2019), Annealing-KD (Jafarpour et al., 2021), RCO (Jin et al., 2019), regular KD, and student without teacher on CIFAR-100 dataset with both ResNet and CNN models

Model	Type	Training method	Accuracy
	teacher(110)	from scratch	71.92
	TA(20)	KD	67.6
ResNet	student(8)	from scratch	61.37
	student(8)	KD	61.41
	student(8)	RCO	61.62
	student(8)	TAKD	61.82
	student(8)	Annealing KD	63.1
	student(8)	**Pro-KD (ours)**	**63.43**

equivalent to making the training process of student easier and smoother. It has been shown in (Li et al., 2017; Chaudhari et al., 2019) that learning a smoother loss is easier than a sharp one. Therefore, we tried to make the training of the student smoother by following the teacher's training steps and also applying the adaptive temperature factor. In addition to this justification, we present a theoretical justification in the Appendix.

5 Experiments and Results

In this section, we evaluate our Pro-KD on 3 different sets of experiments: on image classification, natural language understanding and question answering tasks. In all these three experiments, we compare Pro-KD with the state-of-the-art KD techniques which can address the capacity gap problem such as TAKD (Mirzadeh et al., 2019), and Annealing-KD (Jafari et al., 2021), or RCO (Jin et al., 2019) technique which does not require searching teacher checkpoints for KD. We also include baselines such as the original KD

method (Hinton et al., 2015) and also the regular training without KD.

5.1 Experimental Setup for Image Classification Tasks

Data We used CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009) datasets for our experiments on the image classification tasks. Both datasets have 60,000 of 32×32 color images distributed into 50,000 training and 10,000 test samples, with 10 classes for CIFAR-10 and 100 classes for CIFAR-100

Setup For these experiments, we used ResNet-8 as the student and resNet-110 as the teacher models. The experimental setups are similar to TAKD method (Mirzadeh et al., 2019). Also for the TAKD baseline, we used ResNet-20 as the TA model. The results of these experiments can be found in table 2 and 3 for comparison. For the baselines, first, the ResNet-110 teacher is trained from scratch on the given datasets and then it is used for training baselines with KD. For TAKD baseline, the original KD method is applied to train the TA network with the teacher network and it is applied to train the student network with TA network. For training the student with the proposed Pro-KD method, we trained the teacher for 160 epochs. For training the student, we used maximum temperature 10 and learning rate 0.1. For every 16 epochs, we decrease the temperature by 1 and $n_i = 1$.

Results As it is shown in Tables 2 and 3, Pro-KD outperforms other baselines for both CIFAR-10 and CIFAR-100 experiments. Also, it is worth mentioning that the Annealing-KD technique is the second best result. RCO and TAKD both perform on-par with the regular KD and training the student from scratch.

5.2 Experimental Setup for Natural Language Understanding Tasks on the GLUE Benchmark

Data We use the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018a), which consists of 9 natural language understanding tasks. The tasks cover textual entailment (RTE and MNLI), question-answer entailment (QNLI), paraphrase (MRPC), question paraphrase (QQP), sentiment (SST-2), textual similarity (STS-B), linguistic acceptability (CoLA), and Winograd Schema (WNLI).

Table 4: Dev set results of training DistilRoBERTa using the RoBERTa$_{Large}$ model on the GLUE benchmark.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	Score
RoBERTa$_{Large}$ (Teacher)	67	85	91.63	92.53	96.21	94.53	91.45	89.94/89.97	88.54
DistilRoBERTa(NoKD)	61.9	69.31	89.85	88.46	91.86	91.31	90.04	84.03/ 83.69	83.32
Vanilla KD	60.97	71.11	90.2	88.86	92.54	91.37	91.64	84.18/84.11	83.85
TAKD	61.15	71.84	89.91	88.94	92.54	91.32	91.7	83.89/84.18	83.93
RCO (Jin et al., 2019)	60.66	72.2	90.56	88.41	91.97	91.09	90.04	88.04/84.18	63.63
Annealing KD	61.67	73.64	90.6	89.01	93.11	91.64	91.5	85.34/84.6	84.52
Pro-KD (Ours)	62.14	73.64	91.9	88.8	92.66	91.47	91.53	84.83/84.84	**84.62**

Table 5: Test set results of DistilRoBERTa trained on the GLUE tasks using the RoBERTa$_{Large}$ teacher model.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	Score
Vanilla KD	54.3	74.1	83.4	85.3	93.1	90.8	80.7	83.6/82.9	80.62
TAKD	53.2	74.2	84.7	85	93.2	91.0	80.7	83.8/83.2	80.69
RCO (Jin et al., 2019)	55.1	73.0	87.85	84.4	93.5	88.9	78.66	83.2/82.4	80.51
Annealing KD	54	73.7	85.95	86.8	93.6	90.8	81.15	83.8/83.9	81.23
Pro-KD (Ours)	55.8	73.6	86.45	86.95	93.4	91.0	81.05	84.6/83.8	**81.56**

Table 6: BERT-Small results for Pro-KD on the GLUE dev set

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	Score
BERT$_{Large}$	61.89	68.96	88.22	89.58	92.89	92.4	90.23	86.1/86.25	83.79
BERT-Small (NoKD)	44.05	64.98	83.75	87.41	88.3	86.49	88.43	78.42/78.57	77.74
Vanilla KD	43.28	64.98	84.96	85.95	88.65	86.75	88.24	78.62 /78.55	77.67
TAKD	43.79	65.7	83.98	86.44	88.88	86.78	88.4	78.78/ 78.64	77.84
RCO (Jin et al., 2019)	44.32	65.7	84.91	85.48	88.99	86.32	87.52	78.35 /78.85	77.73
Annealing KD	45	63.9	87.09	87.04	89.56	86.99	88.58	78.66/78.23	78.33
Pro-KD	42.37	66.79	87.78	87.09	89.91	87.88	88.79	79.18/ 79.17	**78.72**

Table 7: The GLUE leaderboard test results of training BERT-Small from the BERT$_{Large}$ teacher.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	Score
BERT-Small (NoKD)	41.3	62.6	79.7	80.05	89	86.3	78.05	78.3/ 77.6	74.37
Vanilla KD	37.3	63.4	80.55	78.15	90.2	86.5	78.25	78/76.5	73.95
TAKD	38.5	62.3	80.5	79.25	89.7	86.7	78	78.2/76.9	74.06
RCO (Jin et al., 2019)	40.3	61.7	79.75	78.95	90.6	86.4	78.35	78.3/ 77.3	74.23
Annealing KD	38.6	63.1	81.85	80.6	91.2	87.3	78.35	77.8// 77.4	74.83
Pro-KD	39	62.7	82.9	80.45	91.2	87.5	79.15	78.6/78.2	**75.16**

Table 8: DistilRoBERTa results for Pro-KD on SQuAD

KD Method	Squad 1.1	Squad 2.0
Teacher	93.7	87
Vanilla KD	85	73.65
TAKD	85.4	73.8
Pro-KD	**86**	**76**

Table 9: BERT-Small results for Pro-KD on SQuAD

KD Method	Squad 1.1	Squad 2.0
Teacher	90.2	81
Vanilla KD	78.54	61.66
TAKD	78.5	61.66
Pro-KD	**79.7**	**62.76**

Setup We perform experiments with multiple students of varying capacities. In the first experiment (Table 4 and 5), we use RoBERTa-large (24-layers) as teacher, DistilRoBERTa (6-layers) (Sanh et al., 2019) as student, and RoBERTa-base (12-layers) (Liu et al., 2019) as the teacher-assistant for the TAKD baseline. For Pro-KD, we train the teacher for 5 epochs, and for training the student we use a maximum temperature of 5, $n_i = 2$, and the learning rate of 2e-5. For the second experiment (Table 6, 7), we use BERT-large (24-layers) as teacher, BERT-Small (4-layers) (Bhargava et al., 2021) as student, and BERT-base (12-layers) (Devlin et al., 2018) as the teacher-assistant for TAKD. We train the teacher for 7 epochs, and for the student we use a maximum temperature of 7 for all tasks, and $n_i = 1$. For the learning rate, we use 5e-5 for RTE and MRPC, and 2e-5 for all other tasks. Additional details about other hyper-parameters can be found in the appendix.

Table 10: Experimenting the impact of the adaptive temperature factor in Pro-KD using DistilRoBERTa trained with RoBERTa-large on the GLUE benchmark

KD Method	CoLA (8.5k)	RTE (2.5k)	MRPC (3.5k)	STS-B (5.7k)	SST-2 (67k)	QNLI (108k)	QQP (363k)	MNLI (392K) (392k)	WNLI	Avg
Pro-KD	**62.14**	**73.64**	**91.9**	**88.8**	**92.66**	91.47	91.53	84.83/84.84	**57.74**	**81.63**
Pro-KD w/o $\mathcal{T}$	58.66	70.03	91.74	88.0	92.43	**91.76**	**91.55**	**84.95/85.41**	56.33	80.61

Results We present our results in Tables 4, 5 and 6. Tables 4 and 6 compare the performance of Pro-KD with Vanilla KD and TAKD on the GLUE dev set, while Table 5 presents the results of DistilRoBERTa on the test set based on GLUE benchmark's leaderboard. We see that Pro-KD outperforms Vanilla KD and TAKD for both Distil-RoBERTa and BERT-Small models. Even though TAKD is able to improve over Vanilla KD, the performance gain is much smaller compared to Pro-KD, demonstrating Pro-KD's high effectiveness in dealing with the large capacity gap problem.

5.3 Experimental Setup for Question Answering

Data We use the Stanford Question Answering Datasets (SQuAD v1.1 (Rajpurkar et al., 2016) and SQuAD v2.0) which are a collection of 100k crowd-sourced question/answer pairs. Given a question and a passage from Wikipedia containing the answer, the task is to predict the answer text span in the passage. The SQuAD 2.0 task extends the SQuAD 1.1 problem definition by allowing for the possibility that no short answer exists in the provided paragraph.

Setup Similar to the GLUE experiments, we perform experiments on 2 different students but with a couple of less baselines because Annealing KD and RCO do not report any result on SQuAD. In the first experiment (Table 8), we use RoBERTa-large (24-layers) as teacher, DistilRoBERTa (6-layers) as student, and RoBERTa-base (12-layers) as the teacher-assistant for the TAKD baseline. We train the teacher for 3 epochs, and for the student we use a maximum temperature of 3, $n_i = 1$, and a learning rate of 3e-5 with a batch size of 12 for both Squad v1.1 and Squad 2.0. For the second experiment (Table 9), we use BERT-large (24-layers) as teacher, BERT-Small (4-layers) as student, and BERT-base (12-layers) as the teacher-assistant for TAKD. We train the teacher for 3 epochs, and for the student we use a maximum temperature of 3, and $n_i = 1$. and a learning rate of 3e-5 with a batch size of 12. Additional details about other

hyper-parameters can be found in the appendix.

Results We present our results in Tables 8 and 9. We again see that Pro-KD consistently outperforms Vanilla KD and TAKD for both Squad v1.1 and Squad 2.0 tasks.

5.4 Ablation Studies and Further Analysis

The Impact of the Temperature Factor Here we show the impact of having the adaptive temperature factor in our technique. The adaptive temperature helps the student model to be exposed to a softened version of the teacher in earlier stages. In this regard we repeated our experiment on the GLUE benchmark when the student is DistilRoBERTa and the teacher is RoBERTa-large. Table 10 shows the result of Pro-KD with and without the temperature. The results indicate that dropping the adaptive temperature will hamper the performance of our model by about 1% on the average GLUE score.

6 Conclusion

In this paper, we highlighted the importance of the capacity-gap problem in KD. We used the VC-dimension analysis to show that dealing with larger teachers may discount the benefit of using knowledge distillation in training. To address this capacity-gap problem, we introduced our Pro-KD technique. Our technique was based on defining a smoother training journey for the student by following the training footprints of the teacher instead of solely relying on distilling from a mature fully-trained teacher. In other words, in contrast to the regular knowledge distillation recipe for model compression in which the student model learns only from a fix fully-trained teacher, in our Pro-KD method, the student learns even from the training path of the teacher. We believe that following the training footsteps of the teacher can be quite influential in improving the student training and can lead to mitigating the capacity-gap problem in KD as well. We showed our theoretical analysis and justifications to support this idea; moreover, we evaluated our technique using a comprehensive set

of experiments on different tasks such as image classification (CIFAR-10 and CIFAR-100), NLP language understanding tasks of the GLUE benchmark, and question answering (SQuAD 1.1 and 2.0) using BERT-based models and consistently got superior results.

Acknowledgments

References

Prajjwal Bhargava, Aleksandr Drozd, and Anna Rogers. 2021. Generalization in nli: Ways (not) to go beyond simple heuristics.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. 2019. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018.

Yevgen Chebotar and Austin Waters. 2016. Distilling knowledge from ensembles of neural networks for speech recognition. In *INTERSPEECH*.

Jang Hyun Cho and Bharath Hariharan. 2019. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4794–4802.

Woo Suk Choi, Kyoung-Woon On, Yu-Jung Heo, and Byoung-Tak Zhang. 2020. Toward general scene graph: Integration of visual semantic knowledge with entity synset alignment. In *Proceedings of the First Workshop on Advances in Language and Vision Research*, pages 7–11, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Urvashi Khandelwal, Christopher D Manning, and Quoc V Le. 2019. Bam! born-again multi-task networks for natural language understanding. *arXiv preprint arXiv:1907.04829*.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bin Dong, Jikai Hou, Yiping Lu, and Zhihua Zhang. 2019. Distillation ≈ Early stopping? harvesting dark knowledge utilizing anisotropic information retrieval for overparameterized neural network. *arXiv preprint arXiv:1910.01255*.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. 2018. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*.

Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born again neural networks. *arXiv preprint arXiv:1805.04770*.

Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. 2020. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029.

Sangchul Hahn and Heeyoul Choi. 2019. Self-knowledge distillation in natural language processing. *arXiv preprint arXiv:1908.01851*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Arthur Jacot, Franck Gabriel, and Clément Hongler. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *arXiv preprint arXiv:1806.07572*.

Aref Jafari, Mehdi Rezagholizadeh, Pranav Sharma, and Ali Ghodsi. 2021. Annealing knowledge distillation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2493–2504, Online. Association for Computational Linguistics.

Borna Jafarpour, Dawn Sepehr, and Nick Pogrebnyakov. 2021. Active curriculum learning. In *Proceedings of the First Workshop on Interactive Learning for Natural Language Processing*, pages 40–45, Online. Association for Computational Linguistics.

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.

Ehsan Kamalloo, Mehdi Rezagholizadeh, Peyman Passban, and Ali Ghodsi. 2021. Not far away, not so close: Sample efficient nearest neighbour data augmentation via MiniMax. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3522–3533, Online. Association for Computational Linguistics.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*.

Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. 2017. Visualizing the loss landscape of neural nets. *arXiv preprint arXiv:1712.09913*.

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. 2020. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4313–4324. PMLR.

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*.

David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. 2015. Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, and Hassan Ghasemzadeh. 2019. Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393*.

Samet Oymak and Mahdi Soltanolkotabi. 2019. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In *International Conference on Machine Learning*, pages 4951–4960. PMLR.

Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. 2021. ALP-KD: attention-based layer projection for knowledge distillation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13657–13665. AAAI Press.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Ahmad Rashid, Vasileios Lioutas, Abbas Ghaddar, and Mehdi Rezagholizadeh. 2020. Towards zero-shot knowledge distillation for natural language processing. *arXiv preprint arXiv:2012.15495*.

Ahmad Rashid, Vasileios Lioutas, and Mehdi Rezagholizadeh. 2021. MATE-KD: Masked adversarial TExt, a companion to knowledge distillation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1062–1071, Online. Association for Computational Linguistics.

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

Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. 2019. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020.

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. Mobilebert: Task-agnostic compression of bert for resource limited devices.

Naftali Tishby and Noga Zaslavsky. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE.

Vladimir N Vapnik. 1998. Statistical learning theory. *Adaptive and learning systems for signal processing communications and control*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. 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.

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

Yimeng Wu, Peyman Passban, Mehdi Rezagholizadeh, and Qun Liu. 2020. Why skip if you can combine: A simple knowledge distillation technique for intermediate layers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1016–1021, Online. Association for Computational Linguistics.

Zhenchuan Yang, Chun Zhang, Weibin Zhang, Jianxiu Jin, and Dongpeng Chen. 2019. Essence knowledge distillation for speech recognition. *ArXiv*, abs/1906.10834.

Ji Won Yoon, Hyeonseung Lee, Hyung Yong Kim, Won Ik Cho, and Nam Soo Kim. 2020. Tutornet: Towards flexible knowledge distillation for end-to-end speech recognition. *ArXiv*, abs/2008.00671.

Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. 2020. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13876–13885.

Wei Zeng, Xiaozhe Ren, Teng Su, Hui Wang, Yi Liao, Zhiwei Wang, Xin Jiang, ZhenZhang Yang, Kaisheng Wang, Xiaoda Zhang, et al. 2021. Pangu-α: Large-scale autoregressive pretrained chinese language models with auto-parallel computation. *arXiv preprint arXiv:2104.12369*.

A Why Does Pro-KD Work?

In this section, first we state stopping the training procedure of the teacher at early epochs will improve Knowledge Distillation. Then, we argue that our method will exploit this fact without requiring any effort on finding the epoch at which early stopping the teacher leads to the optimum Knowledge Distillation. Finally, we will conclude that training the student together with the teacher leads to an important advantage that provides the student with the opportunity of exploiting the knowledge of the optimum teacher.

In knowledge distillation, soft labels coming from a pre-trained teacher contains some so-called *dark knowledge* (Dong et al., 2019) which gives a signal about the relative differences between the probabilities of the classes.

It is shown by (Cho and Hariharan, 2019) and (Dong et al., 2019) that early stopping during training of the teacher network improves the performance of Knowledge Distillation significantly; in other words, there is an optimal epoch at which if teacher training is stopped, the resulting teacher will act as a better teacher compared to the ones stopped at earlier or later epochs. Authors of (Cho and Hariharan, 2019) have shown this fact through experiments based on performing KD by using pre-trained teacher networks stopped at different epochs, and they also have indicated that changing the temperature cannot compensate lost Dark Knowledge. Authors of (Dong et al., 2019) have provided mathematical justification showing that neural network learns more useful information faster. They have used the methods proposed in papers (Du et al., 2018; Oymak and Soltanolkotabi, 2019; Li et al., 2020), and the concept of Neural Tangent Kernel introduced by (Jacot et al., 2018) in order to reach an asymptotic conclusion. This conclusion states that for infinite wide neural networks, gradient descend algorithm searches over different direction with different pace; that is to say, the projection of the loss function in different eigenspaces evolves with different rate. Those rates can be calculated as follows (Dong et al., 2019):

$$
\begin{aligned}
\langle (u_t - y), e_i \rangle &= \langle (I - \eta H^*)(u_t - y), e_i \rangle \\
&= \langle (u_t - y), (I - \eta H^*)e_i \rangle \\
&= (1 - \eta \lambda_i)\langle (u_t - y), e_i \rangle
\end{aligned}
\tag{6}
$$

where u_t is the output of model, y is the true label, e_i is the eigenvector corresponding to the ith

eigenvalue(λ_i) of the static Gram matrix(H^*) of the network. The gram matrix of neural networks is a function of time in general; however, in infinitely wide neural networks, this matrix will be static and called Neural Tangent Kernel (Jacot et al., 2018). This static $n \times n$ matrix, n is the number of training samples, can be calculated as follows (Dong et al., 2019):

$$H^* = \left(\langle \frac{\partial f(\theta, x_i)}{\partial \theta} . \frac{\partial f(\theta, x_j)}{\partial \theta} \rangle \right)_{i,j} \quad (7)$$

where $f(\theta, \cdot)$ is the output of our model, θ are the parameters of the neural network, and x_i is the ith data sample.

Arguments provided by (Dong et al., 2019) state that a neural network learns more useful information faster than non-principal pieces of information about the input samples; however, it does not explain why continuing training teacher will decrease the level of that "Dark Knowledge". To justify this part, we use results from Tishby, Naftali, and Noga Zaslavsky. "Deep learning and the information bottleneck principle."(Tishby and Zaslavsky, 2015) and Saxe, Andrew M., et al. "On the information bottleneck theory of deep learning." (Saxe et al., 2019). In (Saxe et al., 2019) and (Tishby and Zaslavsky, 2015), authors show that there are two stages during training a deep neural network, initial fitting phase and compression phase. These two stages have some important characteristics:

1. Initial Fitting: During this phase, the mutual information between the output of different layers and true labels is increasing. Also, the mutual information between output of layers and the input samples is increasing as well. In other words, network is gaining information about both labels and input samples. Therefore, network is gaining more aforementioned Dark Knowledge as it is gaining information about relative similarities between samples from different classes.

2. Compression phase: During this phase, the mutual information between the output of different layers and true labels is increasing; however, the mutual information between output of layers and the input samples is decreasing. In other words, network tries to compress and discard information which it has gained about input samples (Saxe et al., 2019). We can state that Dark Knowledge is decreasing

during this phase; that is because, network is forgetting relationships between data samples, and at the same time, it is gaining more information about labels. This procedure leads to more confidence on found probabilities for each class. As a result, the information about relative similarities between classes decreases and probabilities tend to one-hot vectors.

Based on the aforementioned facts, a teacher which has stopped learning at the optimum epoch provides more informative pieces of information to the student. Our method exploits the Dark Knowledge of the best teacher by training teacher and students together; in our method, during training teacher and student together, at some point teacher will have the highest level of Dark Knowledge which leads to the best student. The important point here is that we do not need to find the optimum epoch at which this highest level of Dark Knowledge will be achieved. Since we are performing Knowledge Distillation in fixed intervals and storing intermediate checkpoints, the student trained by guidance of the best teacher will be one of these checkpoints and we have access to that.

In contrast with usual KD methods, pre-trained teacher networks are not used in our methods; in fact, we train teacher alongside with the student. Based on the aforementioned arguments, Pro-KD outperforms other KD methods as it gives us the opportunity of exploiting the experience and the Dark Knowledge provided by the best teacher, the teacher which has stopped learning at the optimum epoch.

B Hyper-parameters

In this section, we summarize the hyper-parameters used in our experiments.

Table 11: Model specific Hyper-parameters for BERT-Small on GLUE

Hyper-parameter	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI
Learning Rate	2e-5	5e-5	5e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5
N (Teacher Epochs)	7	7	7	7	7	7	7	7	7
τ_{max}	7	7	7	7	7	7	7	7	7
n (Phase 2 Epochs)	10	10	10	10	10	10	10	10	10

Table 12: Common Hyper-parameters for DistilRoBERTa and BERT-Small models on GLUE

Hyper-parameter	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI
Batch Size	32	32	32	32	32	32	32	32	32
Max Seq. Length	128	128	128	128	128	128	128	128	128
Vanilla KD Alpha	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
Gradient Clipping	1	1	1	1	1	1	1	1	1

Table 13: Model specific Hyper-parameters for DistilRoBERTa on GLUE

Hyper-parameter	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI
Learning Rate	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5
N (Teacher Epochs)	5	5	5	5	5	5	5	5	5
τ_{max}	5	5	5	5	5	5	5	5	5
n (Phase 2 epochs)	10	10	10	10	10	10	10	10	10

Table 14: Model specific Hyper-parameters for Distil-RoBERTa on SQuAD

Hyper-parameter	SQuAD 1.1/2.0	SQuAD Teacher
Learning Rate	3e-5	1.5e-5
Batch Size	12	12
Max Seq. Length	384	384
Doc Stride	128	128
Weight Decay	-	0.01
N (Teacher Epochs)	3	3
τ_{max}	3	-
n (Phase 2 Epochs)	6	-

Table 15: Model specific Hyper-parameters for BERT-Small on SQuAD

Hyper-parameter	SQuAD 1.1/2.0	SQuAD Teacher
Learning Rate	3e-5	3e-5
Batch Size	12	12
Max Seq. Length	384	384
Doc Stride	128	128
N (Teacher Epochs)	3	3
τ_{max}	3	-
n (Phase 2 Epochs)	5	-