Annealing Knowledge Distillation

[1,2]Aref Jafari, [2]Mehdi Rezagholizadeh, [1]Pranav Sharma, [1,3]Ali Ghodsi

[1] David R. Cheriton School of Computer Science, University of Waterloo
[2]Huawei Noah's Ark Lab
[3]Department of Statistics and Actuarial Science, University of Waterloo
{aref.jafari, p68sharma, ali.ghodsi}@uwaterloo.ca
mehdi.rezagholizadeh@huawei.com

Abstract

Significant memory and computational requirements of large deep neural networks restrict their application on edge devices. Knowledge distillation (KD) is a prominent model compression technique for deep neural networks in which the knowledge of a trained large teacher model is transferred to a smaller student model. The success of knowledge distillation is mainly attributed to its training objective function, which exploits the soft-target information (also known as "dark knowledge") besides the given regular hard labels in a training set. However, it is shown in the literature that the larger the gap between the teacher and the student networks, the more difficult is their training using knowledge distillation. To address this shortcoming, we propose an improved knowledge distillation method (called Annealing-KD) by feeding the rich information provided by the teacher's soft-targets incrementally and more efficiently. Our Annealing-KD technique is based on a gradual transition over annealed soft-targets generated by the teacher at different temperatures in an iterative process, and therefore, the student is trained to follow the annealed teacher output in a step-by-step manner. This paper includes theoretical and empirical evidence as well as practical experiments to support the effectiveness of our Annealing-KD method. We did a comprehensive set of experiments on different tasks such as image classification (CIFAR-10 and 100) and NLP language inference with BERT-based models on the GLUE benchmark and consistently got superior results.

1 Introduction

Despite the great success of deep neural networks in many challenging tasks such as natural language processing (Vaswani et al., 2017; Liu et al., 2019), computer vision (Wong et al., 2019; Howard et al., 2017), and speech processing (Chan et al., 2016; He et al., 2019), these state-of-the-art networks are usually heavy to be deployed on edge devices with limited computational power (Bie et al., 2019; Lioutas et al., 2019). A case in point is the BERT model (Devlin et al., 2018) which can be comprised of more than a hundred million parameters.

The problem of network over-parameterization and expensive computational complexity of deep networks can be addressed by neural model compression. There are abundant of neural model compression techniques in the literature (Prato et al., 2019; Tjandra et al., 2018; Jacob et al., 2018), among which knowledge distillation (KD) is one of the most prominent techniques (Hinton et al., 2015). KD is tailored a lot to serve different applications and different network architectures (Furlanello et al., 2018; Gou et al., 2020). For instance, patient KD (Sun et al., 2019), TinyBERT (Jiao et al., 2019), and MobileBERT (Sun et al., 2020) are designed particularly for distilling the knowledge of BERT-based teachers to a smaller student.

The success of KD is mainly attributed to its training objective function, which exploits the soft-target information (also known as "dark knowledge") besides the given regular hard labels in the training set (Hinton, 2012). Previous studies in the literature (Lopez-Paz et al., 2015; Mirzadeh et al., 2019) show that when the gap between the student and teacher models increases, training models with KD becomes more difficult. We refer to this problem as KD's *capacity gap problem* in this paper. For example, Mirzadeh et al. (2019) show that if we gradually increase the capacity of the teacher, first the performance of student model improves for a while, but after a certain point, it starts to drop. Therefore, although increasing the capacity of a teacher network usually boosts its performance, it does not necessarily lead to a better teacher for the student network in KD. In other words, it would

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 2493–2504
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

be more difficult for KD to transfer the knowledge of this enhanced teacher to the student. A similar scenario happens when originally the gap between the teacher and student network is large.

Mirzadeh et al. (2019) proposed their TAKD solution to this problem which makes the KD process more smooth by filling the gap between the teacher and student networks using an intermediate auxiliary network (referred to as "teacher assistant"). The size of this TA network is between the size of the student and the teacher; and it is trained by the teacher first. Then, the student is trained using KD when the TA network is playing the role of its teacher. This way, the training gap (between the teacher and the student) would be less significant compared to the original KD. However, TAKD suffers from the high computational complexity demand since it requires training the TA network separately. Moreover, the training error of the TA network can be propagated to the student during the KD training process.

In this paper, we want to solve the KD *capacity gap problem* from a different perspective. We propose our Annealing-KD technique to bridges the gap between the student and teacher models by introducing a new KD loss with a dynamic temperature term. This way, Annealing-KD is able to transfer the knowledge of the teacher smoothly to the student model via a gradual transition over soft-labels generated by the teacher at different temperatures. We can summarize the contributions of this paper in the following:

1. We propose our novel Annealing-KD solution to the KD *capacity gap problem* based on modifying the KD loss and also introducing a dynamic temperature function to make the student training gradual and smooth.

2. We provide a theoretical and empirical justification for our Annealing-KD approach.

3. We apply our technique to ResNET8 and plain CNN models 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 achieved the-state-of-the art results.

4. Our technique is simple, architecture agnostic, and can be applied on top of different variants of KD.

2 Related Work

2.1 Knowledge Distillation

In the original Knowledge distillation method by Hinton et al. (2015), which is referred to as KD in this paper, the student network is trained based on two guiding signals: first, the training dataset or *hard labels*, and second, the teacher network predictions, which is known as *soft labels*. Therefore, KD is trained based on a linear combination of two loss functions: the regular cross entropy loss function between the student outputs and hard labels, and the KD loss function to minimize the distance between the output predictions of the teacher and student networks at a particular temperature, $\mathcal{T}$, on training samples:

$$\begin{aligned}
&\mathcal{L} = (1-\lambda)\mathcal{L}_{CE} + \lambda\mathcal{L}_{KD} \\
&\mathcal{L}_{CE} = H_{CE}\Big(y, (\sigma(z_s(x))\Big) \\
&\mathcal{L}_{KD} = \mathcal{T}^2 KL\Big(\sigma(\frac{z_t(x)}{\mathcal{T}}), \sigma(\frac{z_s(x)}{\mathcal{T}})\Big)
\end{aligned} \tag{1}$$

where $H_{CE}(.)$ and $KL(.)$ are representing the cross entropy and KL divergence respectively, $z_s(x)$ and $z_t(x)$ are the output logits from the student and teacher networks, $\mathcal{T}$ is the temperature parameter, $\sigma(.)$ is the softmax function and λ is a coefficient between [0,1] to control the contribution of the two loss functions. The above loss function minimizes the distance between the student model and both the underlying function and the teacher model assuming the teacher is a good approximation of the underlying function of the data.

A particular problem with KD, that we would like to address in this paper, is that the larger the gap between the teacher and the student networks, the more difficult is their training using knowledge distillation (Lopez-Paz et al., 2015; Mirzadeh et al., 2019).

2.2 Teacher Assistant Knowledge Distillation (TAKD)

To address the capacity gap problem between the student and teacher networks in knowledge distillation, TAKD (Mirzadeh et al., 2019) proposes to train the student (of small capacity) with a pretrained intermediate network (of moderate capacity) called teacher assistance. In this regard, we first train the TA with the guidance of the teacher network by using the KD method. Then, we can use the learned TA network to train the student network. Here, since the capacity of the TA network is between the capacity of the teacher and the student networks, therefore it can fill the gap between the teacher and student and enhance the complexity of the teacher and transfer its knowledge to the student network.

As it is mentioned in (Mirzadeh et al., 2019), a better idea could be using TAKD in a hierarchical way. So in this case, we can have several TAs with different levels of capacity from large capacities close to the teacher model to small capacities close to the student model. Then we could train these TAs consecutively from large capacities to small capacities in order to have a more smooth transfer of teacher's knowledge to the student model. But it will be difficult. Because, first, since we need to train a new model each time, it is computationally expensive. Second, in this way we will have additive error in each step. Each TA after training will have an approximation error and these errors will accumulate and transfer to the next TA. In the next section, we will propose a simple method to realize this idea and avoid the mentioned problems.

2.3 Annealing in Knowledge Distillation

Clark et al. (2019) proposed an annealing idea in their Born-Again Multi-task (BAM) paper , to train a multitask student network using distillation from some single-task teachers. They introduce a so-called teacher annealing scheme to distill from a dynamic weighted mixture of the teacher prediction and the ground-truth label. In this regard, the weight of teacher's prediction is gradually reduced compared to the weight of ground-truth labels during training. Therefore, early in training, the student model mostly learns from the teacher and later on, it learns mostly from target labels. However, our Annealing-KD is different from Clark et al. (2019) in different aspects. First, the introduced annealing term in BAM is conceptually different from our annealing. While in BAM, teacher annealing controls the contribution of the teacher dark knowledge compared to the ground-truth labels during training, our Annealing-KD is only applied to the teacher output in the KD loss to solve the capacity gap problem between the teacher and student networks. Second, the way we do annealing in our technique is through the temperature parameter and not by controlling the contribution of the teacher and ground-truth labels. Third, BAM falls into another category of knowledge distillation which focuses on improving the performance of the student model and not compressing it. Our method is described in the next section.

3 Method: Annealing Knowledge Distillation

In this section, we describe our Annealing-KD technique and show the rationale behind it. First, we start by formulating the problem and visualizing our technique using an example for a better presentation. Then, we use VC-dimension theory to understand why our technique improves knowledge distillation. We wrap up this section by visualizing the loss landscape of Annealing KD for a ResNet network in order to investigate the impact of our method on the KD loss function.

KD defines a two-objective loss function (i.e. the $\mathcal{L}_{KD}$ and $\mathcal{L}_{CE}$ terms in Equation 1) to minimize the distance between student predictions and soft labels and hard labels simultaneously. Without adding to the computational needs of the KD algorithm, our Annealing-KD model breaks the KD training into two stages: Stage I, gradually training the student to mimic the teacher using our Annealing-KD loss $\mathcal{L}_{\text{KD}}^{\text{Annealing}}$; Stage II, fine-tuning the student with hard labels using $\mathcal{L}_{CE}$. We can define the loss function of our method as following.

$$\mathcal{L} = \begin{cases} \mathcal{L}_{\text{KD}}^{\text{Annealing}}(i), & \textbf{Stage I: } 1 \leq \mathcal{T}_i \leq \tau_{\text{max}} \\ \mathcal{L}_{CE}, & \textbf{Stage II: } \mathcal{T}_n = 1 \end{cases} \tag{2}$$

In the above equation, i indicates the epoch index in the training process with the max epoch number of n for stage I, $\mathcal{T}_i$ represents the temperature value at i^{th} epoch, $\mathcal{L}_{CE}$ is unchanged from Equation 1, and at each epoch (i), $\mathcal{L}_{\text{KD}}^{\text{Annealing}}(i)$ is defined as following:

$$\begin{gathered} \mathcal{L}_{\text{KD}}^{\text{Annealing}}(i) = ||z_s(x) - z_t(x) \times \Phi(\mathcal{T}_i)||_2^2 \\ \Phi(\mathcal{T}) = 1 - \frac{\mathcal{T} - 1}{\tau_{\text{max}}}, 1 \leq \mathcal{T} \leq \tau_{\text{max}}, \mathcal{T} \in \mathbb{N} \end{gathered} \tag{3}$$

In Equation 2, $\mathcal{L}_{\text{KD}}^{\text{Annealing}}$ is defined as an MSE loss between the logits of the student ($z_s(x)$) and an annealed version of the teacher logits ($z_t(x)$), obtained by multiplying the logits by the *annealing function* $\Phi(\mathcal{T})$. The annealing function $\Phi(\mathcal{T})$ can be replaced with any monotonically decreasing function $\Phi : [1, \tau_{\text{max}}] \in \mathbb{N} \rightarrow [0, 1] \in \mathbb{R}$. In stage I of our training, initially we set $\mathcal{T}_1 = \tau_{\max}$ (which leads to the most softened version of the teacher outputs because $\Phi(\mathcal{T}_1) = \frac{1}{\tau_{\max}}$) and decrease the temperature during training as the epoch number grows (that is $\mathcal{T} \rightarrow 1$ while $i \rightarrow n$). Training in stage I continues until $i = n, \mathcal{T} = 1$, for which $\Phi(\mathcal{T}_n) = 1$ and we get the sharpest version of z_t without any softening. The intuition behind using the MSE loss in stage I is that matching the logits of the teacher and student models is a regression task and MSE is one of the best loss functions for this task. We also did an ablation study to compare the performance of MSE and KL-divergence loss function in stage I, and the results of this study support our intuition. For more details, please refer to table 10 of the appendices.

Therefore, our Annealing-KD bridges the gap between the student and teacher models by introducing the dynamic temperature term (that is the annealing function $\Phi(\mathcal{T})$) in the stage I of training. This way our Annealing-KD method is able to smoothly transfer the teacher's knowledge to the student model via a gradual transition over soft-labels generated by the teacher at different temperatures.

To summarize, our Annealing-KD technique is different from KD in following aspects:

- Annealing-KD does not need any λ hyper-parameter to weigh the contribution of the soft and hard lable losses, because it does the training of each loss in a different stage.
- Our Annealing-KD loss $\mathcal{L}_{\text{KD}}^{\text{Annealing}}$ uses $||.||_2^2$ loss instead of the KL divergence.
- Moreover, our technique uses a dynamic temperature by defining the annealing function $\Phi(\mathcal{T})$ in the Annealing-KD loss instead of using a fixed temperature in KD.
- Our empirical experiments showed that it is best to take the network logits instead of the softmax outputs in $\mathcal{L}_{\text{KD}}^{\text{Annealing}}$. Furthermore, in contrast to KD, we do not add the temperature term to student output.

Algorithm 1 explains the proposed method in more detail.

In this section, we proposed an approach to alleviate the gap between the teacher and student models as well as reducing the sharpness of the KD loss function. In our model, instead of pushing the student network to learn a complex teacher function from scratch, we start training the student from a softened version of the teacher and we gradually move toward the original teacher outputs through our annealing process.

Algorithm 1

```
function ANNEALING-KD($S$,$T$,$X$, $k$, $\mathcal{T}_{max}$, $n$)
                                                  ▷ stage I
    for $\mathcal{T} = \tau_{max}$ to 1 do
        $\Phi \leftarrow 1 - \frac{\mathcal{T}-1}{\tau_{\max}}$
        for $i = 1$ to $k$ do
            TRAIN-ANNEALING($S$,$T$, $X$,$\Phi$)
            SAVE-BEST-CHECKPOINT($S$)
        end for
    end for
    $S \leftarrow$ LOAD-BEST-CHECKPOINT  ▷ stage II
    for $i = 1$ to $n$ do
        TRAIN-FINE-TUNE($S$, $X$)
        SAVE-BEST-CHECKPOINT($S$)
    end for
    $S \leftarrow$ LOAD-BEST-CHECKPOINT
    return $S$
end function
```

3.1 Example

For better illustration of our proposed method, we designed a simple example to visualize different parts of our Annealing-KD algorithm. In this regard, we defined a simple regression task using a simple 2D function. This function is a liner combination of three sinusoidal functions with different frequencies $f(x) = sin(3\pi x) + sin(6\pi x) + sin(9\pi x)$. We randomly sample some points from this function to form our dataset (Figure 2-(a)). Next, we fit a simple fully connected neural network with only one hidden layer and the sigmoid activation function to the underlying function of the defined dataset. The teacher model is composed of 100 hidden neurons and trained with the given dataset. After training, the teacher is able to get very close to training data (see the green curve in Figure 2-(a)). We plot the annealed output of the teacher function in 10 different temperatures in Figure. 2-(b). Then, a student model with 10 hidden neurons is trained once with regular KD (Figure 2-(f)) and once with our Annealing-KD (Figures. 2-(c, d, e) depicts the student output at temperatures 10,

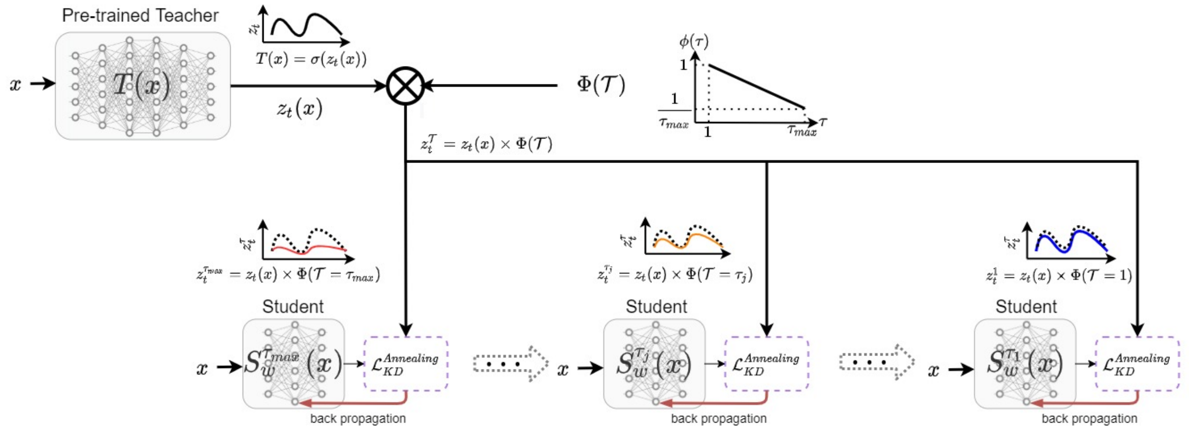

Pre-trained Teacher
x
$T(x)$
$T(x) = \sigma(z_t(x))$
$z_t(x)$
$\Phi(\mathcal{T})$
$\phi(\mathcal{T})$
1
$\frac{1}{\tau_{max}}$
1
τ_{max}
$\mathcal{T}$
$z_t^{\mathcal{T}} = z_t(x) \times \Phi(\mathcal{T})$
$z_t^{\tau_{max}} = z_t(x) \times \Phi(\mathcal{T} = \tau_{max})$
$z_t^{\tau_j} = z_t(x) \times \Phi(\mathcal{T} = \tau_j)$
$z_t^1 = z_t(x) \times \Phi(\mathcal{T} = 1)$
Student
$S_w^{\mathcal{T}_{max}}(x)$
$S_w^{\mathcal{T}_j}(x)$
$S_w^{\mathcal{T}_1}(x)$
$\mathcal{L}_{KD}^{Annealing}$
back propagation

Figure 1: Illustrating the Stage I of the Annealing-KD technique. Given a pre-trained teacher network, we can derive the annealed output of the teacher at different temperature using the annealing function $\Phi(\mathcal{T})$
. We start training of the student from $\mathcal{T} = \tau_{max}$ and go to $\mathcal{T} = 1$.

5, and 1 during the Annealing-KD training). As it is shown in these figures, Annealing-KD guides the student network gradually until it gets to a good approximation of the underlying function and it can match the teacher output better than regular KD.

3.2 Rationale Behind Annealing-KD

Inspired by (Mirzadeh et al., 2020), we can leverage the VC-dimension theory and visulaizing loss landscape to justify why Annealing-KD works better than original KD.

3.2.1 Theoretical Justification

In VC-dimension theory (Vapnik, 1998), the error of classification can be decomposed as:

$$R(f_s) - R(f) \leq O(\frac{|\mathcal{F}_s|_c}{N^{\alpha_s}}) + \varepsilon_s \tag{4}$$

where $R(.)$ is the expected error, $f_s \in \mathcal{F}_s$ is the learner belongs to the function class $\mathcal{F}_s$. f is the underlying function. $|.|_c$ is some function class capacity measure. $O(.)$ is the estimation error of training the learner and ε_s is the approximation error of the best estimator function belonging to the $\mathcal{F}_s$ class (Mirzadeh et al., 2019). Moreover, 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. α is close to $\frac{1}{2}$ for more difficult problems (slow learners) and α is close to 1 for easier problems or fast learners (Lopez-Paz et al., 2015).

In knowledge distillation, we have three main factors: the student (our learner), the teacher, and the underlying function. Based on (Lopez-Paz et al., 2015; Mirzadeh et al., 2019), we can rewrite Equation 4 for knowledge distillation as following:

$$R(f_s) - R(f_t) \leq O(\frac{|\mathcal{F}_s|_c}{n^{\alpha_{st}}}) + \varepsilon_{st} \tag{5}$$

where the student function f_s is following f_t. To define similar inequality for our Annealing-KD technique, we need to consider the effect of the temperature parameter on the three main functions in KD first. For this purpose, we can define $f_s^{\mathcal{T}}$, $f_t^{\mathcal{T}}$, and $f^{\mathcal{T}}$ as the annealed versions of student, teacher, and underlying functions. Furthermore, let $R_{\mathcal{T}}(.)$ to be the expected error function w.r.t the annealed underlying function at temperature $\mathcal{T}$. Hence, for Annealing-KD we have

$$R_{\mathcal{T}}(f_s^{\mathcal{T}}) - R_{\mathcal{T}}(f_t^{\mathcal{T}}) \leq O(\frac{|\mathcal{F}_s|_c}{n^{\alpha_{st}^{\mathcal{T}}}}) + \varepsilon_{st}^{\mathcal{T}}. \tag{6}$$

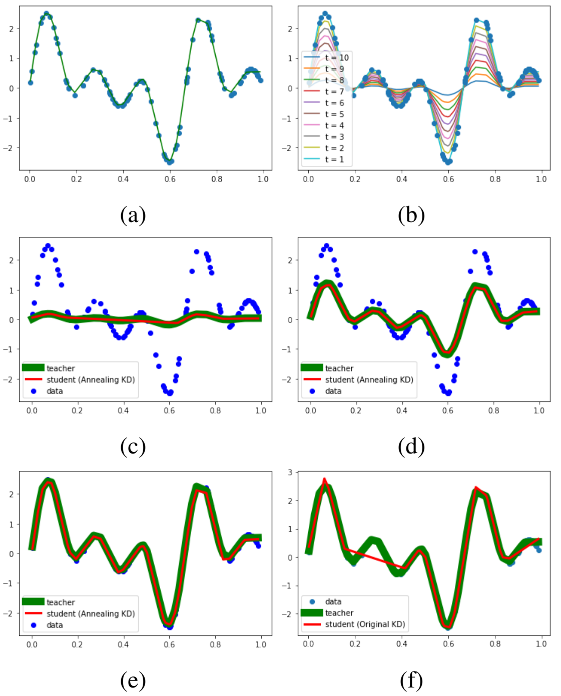
Note that in $\mathcal{T} = 1$, $f_t^1 = f_t$, $f_s^1 = f_s$, $f^1 = f$, and $R_1(.) = R(.)$. Therefore, we can rewrite Equation 6 at $\mathcal{T} = 1$ as:

$$R_1(f_s^1) - R_1(f_t^1) \leq O(\frac{|\mathcal{F}_s|_c}{n^{\alpha_{st}^1}}) + \varepsilon_{st}^1. \tag{7}$$

That being said, to justify that our Annealing-KD is working better than original KD, we can compare Equations 7 and 5 to show the following inequality holds.

$$O(\frac{|\mathcal{F}_s|_c}{n^{\alpha_{st}^1}}) + \varepsilon_{st}^1 \leq O(\frac{|\mathcal{F}_s|_c}{n^{\alpha_{st}}}) + \varepsilon_{st} \tag{8}$$

Since in Annealing-KD, the student network at each temperature is initialized with the trained student network at $f_s^{\mathcal{T}-1}$, the student is much closer to the teacher compared with the original KD method, where the student starts from random a initialization. In other words, in annealing KD, the student



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2: (a) Data samples and trained teacher. (b) Annealed teacher in different temperatures. (c) Student after matching to annealed teacher in $\mathcal{T} = 10$. (d) Student after matching to annealed teacher in $\mathcal{T} = 5$. (e) Student after matching to $\mathcal{T} = 1$. (f) Student trained without KD.

network can learn the annealed teacher at temperature $\mathcal{T}$ faster than the case it starts from a random initial point. Therefore, we can conclude that $\alpha_{st} \leq \alpha_{st}^{\mathcal{T}}$. This property also holds for the last step of annealing KD where $\mathcal{T} = 1$. It means we have $\alpha_{st} \leq \alpha_{st}^{1}$. Furthermore, bear in mind that since the approximation error depends on the capacity of the learner and in annealing KD we do not change the structure of the student, then we expect to have $\varepsilon_{st} = \varepsilon_{st}^{\mathcal{T}}$. Therefore, based on these two evidence ($\alpha_{st} \leq \alpha_{st}^{\mathcal{T}}$ and $\varepsilon_{st} = \varepsilon_{st}^{\mathcal{T}}$), we can conclude that Equation 8 holds.

3.2.2 Empirical Justification

Because of the non-linear nature of neural networks, the loss functions of these models are non-convex. This property might prevent a learner from a good generalization. There are some beliefs in the community of machine learning, this phenomena can be harsher in the sharp loss functions than the flat loss functions (Chaudhari et al., 2019; Hochreiter and Schmidhuber, 1997). Although, there are some arguments around this belief (Li et al., 2018), for the case of knowledge distillation it seems flatter loss functions are related to higher accuracy (Mirzadeh et al., 2019; Zhang et al., 2018; Hinton et al., 2015). One of the advantages of annealing the teacher function during training is reducing the sharpness of annealing loss function in the early steps of stage I. In other words, the sharpness of the loss function in annealing KD changes dynamically. In the early steps of annealing when the temperature is high, the loss function is flatter. This helps the student to train the teacher network's behaviour faster and easier.

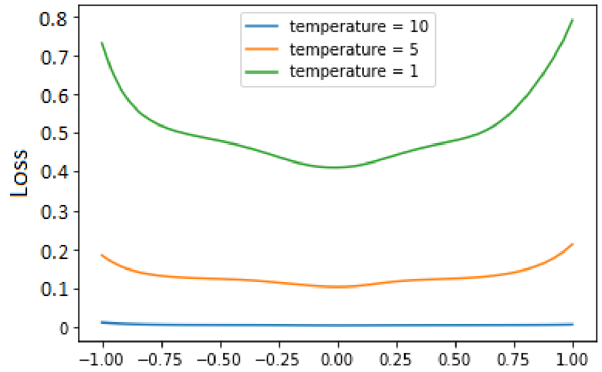


Figure 3: Visualization of annealing KD loss function in stage I for ResNet 8 student during the training on CIFAR-10 dataset in different temperatures

In order to compare the effect of different temperatures, the loss landscape visualization method in (Li et al., 2018) is used to plot the loss behaviour of CIFAR-10 experiment with ResNet 8 student in Figure. 3. Here as it is shown, by decreasing the temperature during the training, the sharpness of the loss function increases. So the student network can avoid many of the bad local minimums in the early stages of the algorithm when the temperature is high. Then in the final stages of the algorithm, when the loss function is sharper, the network starts from a much better initialization.

4 Experiments

In this section, we describe the experimental evaluation of our proposed Annealing KD method. We evaluate our technique on both image classification and natural language inference tasks. In all of our experiments, we compare the annealing KD results with TAKD, standard KD, and training student without KD results.

4.1 Datasets

For image classification, we assess Annealing-KD on CIFAR-10 and CIFAR-100 datasets (Krizhevsky et al., 2009) which are image datasets containing

32 × 32 color images with 10 and 100 classes respectively. For the natural language inference task, we employ the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018), which is a collection of nine different tasks for training, evaluating, and analyzing natural language understanding models. GLUE consists of Multi-Genre Natural Language Inference (MNLI) (Williams et al., 2017), Quora Question Pairs (QQP) (Chen et al., 2018), Question Natural Language Inference (QNLI) (Rajpurkar et al., 2016), Stanford Sentiment Treebank (SST-2) (Socher et al., 2013), Corpus of Linguistic Acceptability (COLA) (Warstadt et al., 2019), Semantic Textual Similarity Benchmark (STS-B) (Cer et al., 2017), Microsoft Research Paraphrase Corpus (MRPC) (Dolan and Brockett, 2005), Recognizing Textual Entailment (RTE) (Bentivogli et al., 2009), Winograd NLI (WNLI) (Levesque et al., 2012).

4.2 Experimental Setup for Image Classification Tasks

For image classification experiments, we used CIFAR-10 and CIFAR-100 datasets with the same experimental setup in the TAKD method (Mirzadeh et al., 2020). In these experiments, we used ResNet and plain CNN networks as the teacher, student, and also the teacher assistant for the TAKD baseline. For the ResNet experiments, we used ResNet-110 as the teacher and ResNet-8 as the student. For plain CNN experiments, we used CNN network with 10 layers as teacher and 2 layers as the student according to TAKD. Also, for the TAKD baseline, we used ResNet-20 and CNN with 4 layers as the teacher assistant. Tables 1 and 2 compare the annealing KD performance with other baselines over CIFAR-10 and CIFAR-100 datasets respectively. For the ResNet experiments in both tables 1 and 2, the teacher ResNet-110 is trained from scratch and a ResNet-20 TA is trained by the teacher using KD. Then we would like to train a ResNet-8 student using different techniques and compare their performance against our Annealing KD method. In this regard, we evaluate the performance of training the student from scratch, training with the large ResNet-110 teacher using KD, training with TA as the teacher and using our Annealing-KD approach. The results of this experiment with ResNet show that our Annealing-KD outperforms all other baselines and TAKD is the second-best performing student without significant distinction compared to KD. More details about the training hyper-parameters are added to the appendix A.

Table 1: Comparing the test accuracy of annealing KD, TAKD, regular KD, and student without teacher on CIFAR-10 dataset with both ResNet and CNN models

Model	Type	Training method	Accuracy
ResNet	Teacher(110)	from scratch	93.8
	TA(20)	KD	92.39
	Student(8)	from scratch	88.44
	Student(8)	KD	88.45
	Student(8)	TAKD	88.47
	Student(8)	**Annealing KD (ours)**	**89.44**
CNN	Teacher(10)	from scratch	90.1
	TA(4)	KD	82.39
	Student(2)	from scratch	72.75
	Student(2)	KD	72.43
	Student(2)	TAKD	72.62
	Student(2)	**Annealing KD (ours)**	**73.17**

Table 2: Comparing the test accuracy of annealing KD, TAKD, regular KD, and student without teacher on CIFAR-100 dataset with both ResNet and CNN models

Model	Type	Training method	Accuracy
ResNet	teacher(110)	from scratch	71.92
	TA(20)	KD	67.6
	student(8)	from scratch	61.37
	student(8)	KD	61.41
	student(8)	TAKD	61.82
	student(8)	**Annealing KD (ours)**	**63.1**
CNN	Teacher(10)	from scratch	64.89
	TA(4)	KD	60.73
	student(2)	from scratch	51.35
	student(2)	KD	51.62
	student(2)	TAKD	51.85
	student(2)	**Annealing KD (ours)**	**53.35**

4.3 Experimental setup for GLUE tasks

For these set of experiments, we use the GLUE benchmark which consists of 9 natural language understanding tasks. In the first experiment (Table 3), 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. For Annealing KD, we use a maximum temperature of 7, learning rate of 2e-5, and train for 14 epochs in phase 1, and 6 epochs in phase 2. In table 3 the Annealing KD and the other baselines performances on dev set of GLUE tasks are compared. Also, we compared the performances of these methods on test set based on the GLUE benchmark's leaderboard results in table 4. In the second experiment (Table 5), we use

Table 3: DistilRoBERTa results for Annealing KD on dev set. F1 scores are reported for MRPC, pearson correlations for STB-B, and accuracy scores for all other tasks.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI	Score
Teacher	68.1	86.3	91.9	92.3	96.4	94.6	91.5	90.22/89.87	56.33	85.29
From scratch	59.3	67.9	88.6	88.5	92.5	90.8	90.9	84/84	52.1	79.3
Vanilla KD	60.97	71.11	90.2	88.86	92.54	91.37	91.64	84.18/84.11	56.33	80.8
TAKD	61.15	71.84	89.91	88.94	92.54	91.32	**91.7**	83.89/84.18	56.33	80.85
Annealing KD	**61.67**	**73.64**	**90.6**	**89.01**	**93.11**	**91.64**	91.5	**85.34/84.6**	56.33	**81.42**

Table 4: Performance of DistilRoBERTa trained by annealing KD on the GLUE leaderboard compared with Vanilla KD and TAKD. We applied the standard tricks to all 3 methods and fine-tune RTE, MRPC and STS-B from trained MNLI student model.

KD Method	CoLA	MRPC	STS-B	SST-2	MNLI-m	MNLI-mm	QNLI	QQP	RTE	WNLI	Score
Vanilla KD	**54.3**	86/80.8	85.7/84.9	93.1	83.6	82.9	90.8	71.9/89.5	74.1	65.1	78.9
TAKD	53.2	86.7/82.7	85.6/84.4	93.2	83.8	83.2	**91**	72/89.4	**74.2**	65.1	79
Annealing KD	54	**88.0/83.9**	**87.0/86.6**	**93.6**	**83.8**	**83.9**	90.8	**72.6/89.7**	73.7	65.1	**79.5**

Table 5: BERT-Small results for Annealing KD on dev set. F1 scores are reported for MRPC, pearson correlations for STS-B, and accuracy scores for all other tasks.

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI	Score
Teacher	65.8	71.48	89.38	89.2	92.77	92.82	91.45	86.3/86.4	60.56	82.19
Vanilla KD	33.5	57	86	72.3	88.76	**83.15**	**87**	72.62/73.19	54.92	70.58
TAKD	34.24	59.56	85.23	71.1	89.1	82.62	87	72.32/72.45	54.92	70.76
Annealing KD	**35.98**	**61**	**86.2**	**74.54**	**89.44**	83.14	86.5	**73.85/74.84**	54.92	**71.68**

BERT-large (24 layers) as teacher, BERT-small (4 layers) as student, and BERT-base (12 layers) as the teacher assistant of TAKD. We use a maximum temperature of 7 for MRPC, SST-2, QNLI, and WNLI, and 14 for all other tasks. The number of epochs in phase 1 is twice the maximum temperature, and 6 in phase 2. We use the learning rate of 2e-5 for all tasks except RTE and MRPC which use 4e-5. Table 5 compares the performance of annealing KD and other baselines on dev set for small-BERT experiments. For more details regarding other hyper-parameters, refer to the appendix. We also perform ablation on the choice of loss function in phase 1, and choice of different max temperature values, both of which can be found in the appendix.

4.4 GLUE Results

We present our results in Tables 3, 4, and 5. We see that Annealing KD consistently outperforms the other techniques both on dev set as well as the GLUE leaderboard. Furthermore, in table 5, when we reduce the size of the student to a 4 layer model (BERT-Small), we notice almost twice as big of a gap in the average score over Vanilla KD when compared with DistilRoBERTa (Table 3). We can also observe TAKD improving slightly over Vanilla KD, with the improvement being more significant in the case of the smaller student (BERT-Small).

5 Discussion

In image classification experiments, the improvement gap between the annealing KD results and the other baselines in CIFAR-100 experiments is larger than CIFAR-10 ones. We can observe similar conditions for the NLP experiments between BERT-small and DistilRoBERTa students (the performance gap of BERT-small is larger). In both of these cases, the problem for the student was more difficult. CIFAR-100 dataset is more complex than CIFAR-10 dataset. So the teacher has learned a more complex function that should be transferred to the student. In NLP experiments, on the other hand, the tasks are the same but BERT-small student has a smaller capacity in compare with DistilRoBERTa. Therefore the problem is more difficult for BERT-small. From this observation, we can conclude, whenever the gap between the teacher and student is larger, the annealing KD performs better than the other baselines and leverage the acquired knowledge by the teacher to train the student.

6 Conclusion and Future Work

In this work, we discussed that the difference between the capacity of the teacher and student models in knowledge distillation may hamper its performance. On the other hand, in most cases, larger neural networks can be trained better and get more accurate results. If we consider better teachers can train better students, then larger teachers with better accuracy would be more favourable for knowledge distillation training. In this paper, we proposed an improved knowledge distillation method called annealing KD to alleviate this problem and leverage the knowledge acquired by more complex teachers to guide the small student models better during their training. This happened by feeding the rich information provided by the teacher's soft-targets incrementally and more efficiently. Our Annealing-KD technique was based on a gradual transition over annealed soft-targets generated by the teacher at different temperatures in an iterative process; and therefore, the student was trained to follow the annealed teacher output in a step-by-step manner.

References

Bentivogli, L., Clark, P., Dagan, I., and Giampiccolo, D. (2009). The fifth pascal recognizing textual entailment challenge. In *TAC*.

Bie, A., Venkitesh, B., Monteiro, J., Haidar, M., Rezagholizadeh, M., et al. (2019). Fully quantizing a simplified transformer for end-to-end speech recognition. *arXiv preprint arXiv:1911.03604*.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964. IEEE.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J., Sagun, L., and Zecchina, R. (2019). Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018.

Chen, Z., Zhang, H., Zhang, X., and Zhao, L. (2018). Quora question pairs.

Clark, K., Luong, M.-T., Khandelwal, U., Manning, C. D., and Le, Q. V. (2019). BAM! born-again multi-task networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5931–5937, Florence, Italy. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dolan, W. B. and Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Furlanello, T., Lipton, Z. C., Tschannen, M., Itti, L., and Anandkumar, A. (2018). Born again neural networks. *arXiv preprint arXiv:1805.04770*.

Gou, J., Yu, B., Maybank, S. J., and Tao, D. (2020). Knowledge distillation: A survey. *arXiv preprint arXiv:2006.05525*.

He, Y., Sainath, T. N., Prabhavalkar, R., McGraw, I., Alvarez, R., Zhao, D., Rybach, D., Kannan, A., Wu, Y., Pang, R., et al. (2019). Streaming end-to-end speech recognition for mobile devices. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6381–6385. IEEE.

Hinton, G. (2012). Neural networks for machine learning, coursera. *URL: http://coursera.org/course/neuralnets*.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the Knowledge in a Neural Network. *arXiv e-prints*, page arXiv:1503.02531.

Hochreiter, S. and Schmidhuber, J. (1997). Flat minima. *Neural Computation*, 9(1):1–42.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., and Kalenichenko, D. (2018). Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2704–2713.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., and Liu, Q. (2019). Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Levesque, H., Davis, E., and Morgenstern, L. (2012). The winograd schema challenge. In *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*. Citeseer.

Li, H., Xu, Z., Taylor, G., Studer, C., and Goldstein, T. (2018). Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399.

Lioutas, V., Rashid, A., Kumar, K., Haidar, M. A., and Rezagholizadeh, M. (2019). Distilled embedding: non-linear embedding factorization using knowledge distillation. *arXiv preprint arXiv:1910.06720*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lopez-Paz, D., Bottou, L., Schölkopf, B., and Vapnik, V. (2015). Unifying distillation and privileged information. *arXiv preprint arXiv:1511.03643*.

Mirzadeh, S., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., and Ghasemzadeh, H. (2020). Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *AAAI 2020*, abs/1902.03393.

Mirzadeh, S.-I., Farajtabar, M., Li, A., and Ghasemzadeh, H. (2019). Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393*.

Prato, G., Charlaix, E., and Rezagholizadeh, M. (2019). Fully quantized transformer for improved translation. *arXiv preprint arXiv:1910.10485*.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Sun, S., Cheng, Y., Gan, Z., and Liu, J. (2019). Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., and Zhou, D. (2020). Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.

Tjandra, A., Sakti, S., and Nakamura, S. (2018). Tensor decomposition for compressing recurrent neural network. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Vapnik, V. (1998). *Statistical learning theory*. John Wiley and Sons.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Warstadt, A., Singh, A., and Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Williams, A., Nangia, N., and Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Wong, A., Famuori, M., Shafiee, M. J., Li, F., Chwyl, B., and Chung, J. (2019). Yolo nano: a highly compact you only look once convolutional neural network for object detection. *arXiv preprint arXiv:1910.01271*.

Zhang, Y., Xiang, T., Hospedales, T. M., and Lu, H. (2018). Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328.

Appendices

A Experimental parameters of the image classification tasks

In this section, we include more detail of our experimental settings of section 4.2 in the paper. For the baseline experiments, we used the same experimental setup as (Mirzadeh et al., 2019). We performed two series of experiments based on ResNet and plain CNN neural networks on CIFAR-10 and CIFAR-100 datasets. Table 6 illustrates the hyper-parameters used in these experiments. (BS = batch size, EP1= number of epochs in phase 1 (for the baselines, this is the number of training epochs), EP2 = number of epochs in phase 2, LR = learning rate, MO = momentum, WD = weight decay, τ_{max} = maximum temperature)

Table 6: Hyper-parameters of CIFAR-10 and CIFAR-100 experiments

Model	Type	Training method	BS	EP1	EP2	LR	MO	WD	$\tau_{\max}$
ResNet	Teacher(110)	from scratch	128	160	N/A	0.1	0.9	10^{-4}	N/A
	TA(20)	KD	128	160	N/A	0.1	0.9	10^{-4}	N/A
	Student(8)	from scratch	128	160	N/A	0.1	0.9	10^{-4}	N/A
	Student(8)	KD	128	160	N/A	0.1	0.9	10^{-4}	1
	Student(8)	TAKD	128	160	N/A	0.1	0.9	10^{-4}	1
	Student(8)	**Annealing KD (ours)**	128	160	160	0.1	0.9	10^{-4}	10
CNN	Teacher(10)	from scratch	128	160	N/A	0.1	0.9	10^{-4}	N/A
	TA(4)	KD	128	160	N/A	0.1	0.9	10^{-4}	N/A
	Student(2)	from scratch	128	160	N/A	0.1	0.9	10^{-4}	N/A
	Student(2)	KD	128	160	N/A	0.1	0.9	10^{-4}	1
	Student(2)	TAKD	128	160	N/A	0.1	0.9	10^{-4}	1
	Student(2)	**Annealing KD (ours)**	128	160	160	0.1	0.9	10^{-4}	10

B BERT Experiments

In these experiments, RoBERTa-large (24 layers) and DistilRoBERTa (6 layers) are used as the teacher and student models respectively. Also, RoBERTa-base (12-layer) is used as the teacher assistant for the TAKD baseline. For Annealing KD, we use the maximum temperature of 7 and the learning rate of 2e-5 for all the tasks. We trained the student model for 14 epochs in phase 1, and 6 epochs in phase 2. Table 8 illustrates the details of the hyper-parameters of the experiments. Also, Table 11 illustrates the hyper-parameter values of BERT-small experiments in detail. Also, we did two ablation studies. In the first one, we tried to fine-tune the maximum temperature in annealing KD and check the performance improvement compared with using the general value of 7. As it is illustrated in Table 9, we can get more improvement with selecting the maximum temperature parameter more carefully. The second ablation is about comparing the effect of mean square error and KL-divergence loss functions on the final results of the experiments when they are used as the loss function of the first phase. Table 10 shows the results of this ablation.

Table 7: Common Hyper-parameters for DistilRoBERTa and BERT-Small models on GLUE tasks

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 8: Model specific Hyper-parameters for DistilRoBERTa on GLUE tasks

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
Phase 1 epochs	14	14	14	14	14	14	14	14	14
Phase 2 epochs	4	4	4	4	4	4	4	4	4
τ_{max}	7	7	7	7	7	7	7	7	7

Table 9: Ablation on DistilRoberta Annealing KD with temperature tuning

KD Method	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI	Avg
Annealing KD	61.67	73.64	90.6	89.01	93.11	91.64	91.5	85.34/84.6	56.33	81.42
+ temp tuning	61.67	73.64	**91.99**	**89.26**	**93.34**	**92**	**91.72**	**85.14/85.22**	56.33	**81.67**
(max temperature)	7	7	8	14	14	11	14	14	7	-

Table 10: Ablation on DistilRoberta Annealing KD with different loss functions

KD Method and Loss	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI	Avg
Annealing KD, MSE	61.67	73.64	90.6	89.01	93.11	91.64	91.5	85.34/84.6	56.33	81.42
Annealing KD, KL-div	62.56	70.75	90.84	89.01	93	91.32	91.42	85/84.75	56.33	81.13

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

Hyper-parameter	CoLA	RTE	MRPC	STS-B	SST-2	QNLI	QQP	MNLI	WNLI
Learning Rate	2e-5	4e-5	4e-5	2e-5	2e-5	2e-5	2e-5	2e-5	2e-5
Phase 1 epochs	28	28	14	28	14	14	28	28	14
Phase 2 epochs	6	6	6	6	6	6	6	6	6
τ_{max}	14	14	7	14	7	7	14	14	7