Asymmetric Temperature Scaling Makes Larger Networks Teach Well Again

Xin-Chun Li[1], **Wen-Shu Fan**[1], **Shaoming Song**[2], **Yinchuan Li**[2]
Bingshuai Li[2], **Yunfeng Shao**[2], **De-Chuan Zhan**[1]
[1] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
[2] Huawei Noah's Ark Lab, Beijing, China
{lixc, fanws}@lamda.nju.edu.cn, zhandc@nju.edu.cn
{shaoming.song, liyinchuan, libingshuai, shaoyunfeng}@huawei.com

Abstract

Knowledge Distillation (KD) aims at transferring the knowledge of a well-performed neural network (the *teacher*) to a weaker one (the *student*). A peculiar phenomenon is that a more accurate model doesn't necessarily teach better, and temperature adjustment can neither alleviate the mismatched capacity. To explain this, we decompose the efficacy of KD into three parts: *correct guidance*, *smooth regularization*, and *class discriminability*. The last term describes the distinctness of *wrong class probabilities* that the teacher provides in KD. Complex teachers tend to be over-confident and traditional temperature scaling limits the efficacy of *class discriminability*, resulting in less discriminative wrong class probabilities. Therefore, we propose *Asymmetric Temperature Scaling (ATS)*, which separately applies a higher/lower temperature to the correct/wrong class. ATS enlarges the variance of wrong class probabilities in the teacher's label and makes the students grasp the absolute affinities of wrong classes to the target class as discriminative as possible. Both theoretical analysis and extensive experimental results demonstrate the effectiveness of ATS.

1 Introduction

Although large-scale deep neural networks have achieved overwhelming successes in many real-world applications [22, 11, 60], the vast capacity hinders them from being deployed on portable devices with limited computation and storage resources [3]. Some efficient architectures, e.g., MobileNets [14, 37] and ShuffleNets [59, 29], have been proposed for lightweight deployment, while their performances are usually constrained. Fortunately, knowledge distillation (KD) [46, 13] could transfer the knowledge of a more complex and well-performed network (i.e., the *teacher*) to them.

The original KD [13] forces the student to mimic the teacher's behavior via minimizing the Kullback-Leibler (KL) divergence between their output probabilities. Recent studies generalize KD to various types of knowledge [36, 57, 17, 12, 33, 1, 34, 44, 52, 27, 54, 45, 50, 26, 23] or various distillation schemes [61, 2, 58, 20]. An intuitive sense after the proposal of KD [13] is that larger teachers could teach students better because their accuracies are higher. A recent work [6] first points out that the teacher accuracy is a poor predictor of the student's performance. That is, more accurate neural networks don't necessarily teach better. Until now, this phenomenon is still counter-intuitive [51], surprising [31], and unexplored [24]. Different from some existing empirical studies and theoretical analysis [40, 18, 30, 35, 55, 63, 6, 28, 15], we investigate the miraculous phenomenon in detail and aim to answer the following questions: *What's the real reason that more complex teachers can't teach well? Is it really impossible to make larger teachers teach better through simple operations, such as temperature scaling?*

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

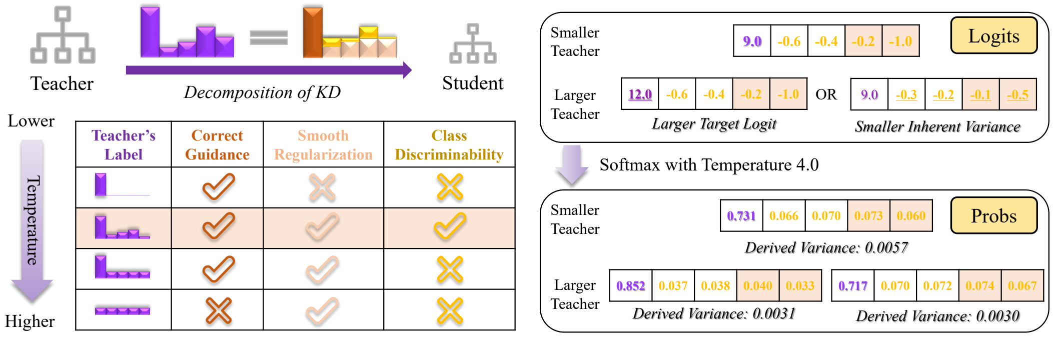

Figure 1: **Left**: Decomposition of a teacher's label. The first class is the target. As temperature increases, *correct guidance* is weaker, *smooth regularization* is stronger, while *class discriminability* (measured by the variance of wrong class probabilities) will first increase and then decrease. **Right**: Larger/Smaller teachers' logits are consistent in relative class affinities, i.e., logit values of the four wrong classes are in the same order of magnitude. However, larger teachers are over-confident and give a larger target logit or smaller *inherent variance*, leading to a smaller *derived variance* under traditional temperature scaling, i.e., less distinct wrong class probabilities after softmax.

To answer the first question, we focus on analyzing the distinctness of wrong class probabilities that a teacher provides in KD. We decompose the teacher's label into three parts (see Sect. 4.1): (I) *Correct Guidance*: the correct class's probability; (II) *Smooth Regularization*: the average probability of wrong classes; (III) *Class Discriminability*: the variance of wrong class probabilities (defined as *derived variance*). The commonly utilized temperature scaling could control the efficacy of these three terms (the left of Fig. 1). More complex teachers are over-confident and assign a larger score for the correct class or less varied scores for the wrong classes. If we use a uniform temperature to scale their logits, the *class discriminability* of the larger teacher is less effective (theoretically analyzed in Sect. 4.2), i.e., the probabilities of wrong classes are less distinct (the right of Fig. 1).

As to the second question, we focus on enlarging the variance of wrong class probabilities (i.e., *derived variance*) that a teacher provides to make the distillation process more discriminative. To specifically enhance the distinctness of wrong class probabilities, we separately apply a higher/lower temperature to the correct/wrong class's logit instead of a uniform temperature (see Sect. 4.3). We name our method *Asymmetric Temperature Scaling (ATS)*, and abundant experimental studies verify that utilizing this simple operation could make larger teachers teach well again.

2 Related Works

KD with Larger Teacher: Although KD has been a general technique for knowledge transfer in various applications [13, 61, 42, 25], could any student learn from any teacher? [6] first studies the KD's dependence on student and teacher architectures. They find that larger models do not often make better teachers and propose the *early-stopped teacher* as a solution. [31] introduces a multi-step KD process, employing an intermediate-sized network (the *teacher assistant*) to bridge the capacity gap. [51] formulates KD as a multi-task learning problem with several knowledge transfer losses. The transfer loss will be utilized only when its gradient direction is consistent with the cross-entropy loss. [10, 24] define the knowledge gap as *residual*, which is utilized to teach the *residual student*, and then they take the ensemble of the student and residual student for inference. These works attribute the worse teaching performance to capacity mismatch, i.e., weaker students can't completely mimic the excellent teachers. However, they don't explain this peculiar phenomenon in detail.

Understanding of KD: Quite a few works focus on understanding the advantages of KD from a principled perspective. [28] unifies KD and privileged information into *generalized distillation*. [35, 18] utilize gradient flow and neural tangent kernel to analyze the convergence property of KD under deep linear networks and infinitely wide networks. [5] explains KD via quantifying the task-relevant and task-irrelevant visual concepts. [7] casts KD as a semiparametric inference problem and proposes corresponding enhancements. Our work is more related to KD decompositions. [9] treats the teacher's correct/wrong outputs differently, respectively explaining them as importance weighting

Table 1: The used notations in this paper. The definitions of *Derived Average*, *Derived Variance* and *Inherent Variance* are only for wrong classes (Sect. 4.1 and Sect. 4.2).

	All Classes	Wrong Classes
Logit	$\mathbf{f}$	$\mathbf{g} = [\mathbf{f}_c]_{c \neq y}$
Probability	$\mathbf{p} = \text{SF}(\mathbf{f})$	$\mathbf{q} = [\mathbf{p}_c]_{c \neq y}$
Derived Average of Probabilities	-	$e(\mathbf{q}) = \sum_j \mathbf{q}_j / (C-1)$
Derived Variance of Probabilities	-	$v(\mathbf{q}) = \sum_j (\mathbf{q}_j - e(\mathbf{q}))^2 / (C-1)$
Inherent Variance of Probabilities	-	$\tilde{\mathbf{q}} = \text{SF}(\mathbf{g})$, $v(\tilde{\mathbf{q}}) = \sum_j (\tilde{\mathbf{q}}_j - e(\tilde{\mathbf{q}}))^2 / (C-1)$

and class similarities. [40] further decomposes the "dark knowledge" into universal knowledge, domain knowledge, and gradient rescaling. [30] establishes a bias-variance tradeoff to quantify the divergence of a teacher with the *Bayes teacher*. [63] utilizes bias-variance decomposition to analyze KD and discovers *regularization samples* that could increase bias and decrease variance. Our work is also related to Label smoothing (LS). [55] points out that the regularization effect in KD is similar to LS. [32] finds that training a teacher with LS could degrade its teaching quality, and attributes this to the fact that LS erases *relative information* between teacher logits. Recently, [38] further studies this problem and proposes a metric to measure the degree of erased information quantitatively. Our work also decomposes KD into several effects to study why more complex teachers can't teach well. Detailed relatedness to these works is presented in Sect. 4.1.

3 Background

We consider a C-class classification problem with $\mathcal{Y} = [C] = \{1, 2, \cdots, C\}$. Given a neural network and a sample pair $(\mathbf{x}, y)$, we could obtain the "logits" as $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^C$. We denote the softmax function with temperature τ as SF$(\cdot; \tau)$, i.e., $\mathbf{p}_c(\tau) = \exp\left(\mathbf{f}_c(\mathbf{x})/\tau\right)/Z(\tau)$ and $Z(\tau) = \sum_{j=1}^{C} \exp\left(\mathbf{f}_j(\mathbf{x})/\tau\right)$, where $\mathbf{p}(\tau)$ is the softened probability vector that a network outputs and c is the index of class. Later, we may omit the dependence on $\mathbf{x}$ and τ if without any ambiguity. We use $\mathbf{f}_y$ and $\mathbf{p}_y$ to denote the *correct class*'s logit and probability, while we use $\mathbf{g}$ and $\mathbf{q}$ to represent the vector of *wrong classes*' logits and probabilities, i.e., $\mathbf{g} = [\mathbf{f}_c]_{c \neq y}$ and $\mathbf{q} = [\mathbf{p}_c]_{c \neq y}$. The notations could be found in Tab. 1.

The most standard KD [13] contains two stages of training. The first stage trains complex teachers, and then the second stage transfers the knowledge from teachers to a smaller student via minimizing the KL divergence between softened probabilities. Usually, the loss function during the second stage (i.e., the student's learning objective) is a combination of cross-entropy loss and distillation loss:

$$\ell = \underbrace{-(1-\lambda)\log \mathbf{p}_y^S(1)}_{\text{CE Loss}} \underbrace{-\lambda\tau^2 \sum_{c=1}^{C} \mathbf{p}_c^T(\tau) \log \mathbf{p}_c^S(\tau)}_{\text{KD Loss}}, \tag{1}$$

where the upper script "T"/"S" denotes "Teacher"/"Student" respectively. Commonly, a default temperature of 1 is utilized for the CE loss, and the student could also take a temperature of 1 for the KD loss, e.g., $\mathbf{p}_c^S(\tau = 1)$ [13, 31, 45, 44].

Suppose we have two teachers, denoted as T_{large} and T_{small}, and the larger teacher performs better on both training and test data. If we use them to teach the same student S, we could find that the student's performance is worse when mimicking the larger teacher's outputs. Adjusting the temperature could neither make the larger teacher teach well. The details of this phenomenon could be found in [6, 31] and Fig. 9. Obviously, $\mathbf{p}^{T_{\text{large}}}$ could differ a lot from $\mathbf{p}^{T_{\text{small}}}$, which is the only difference in the loss function when teaching the student. Hence, we focus on analyzing *what probability distributions are tended to be provided by teachers with different capacities*.

4 Proposed Methods

This section first decomposes KD into three parts and defines several quantitative metrics. Then, we present theoretical analysis to demonstrate why larger networks can't teach well. Finally, we propose a more appropriate temperature scaling approach as an alternative.

4.1 KD Decomposition

We omit the coefficient of $\lambda\tau^2$ in Eq. 1, and define $e\left(\mathbf{q}^T(\tau)\right) = \frac{1}{C-1}\sum_{j=1,j\neq y}^{C}\mathbf{p}_j^T(\tau)$, where $\mathbf{q}^T(\tau) = [\mathbf{p}_c^T(\tau)]_{c\neq y}$. Then, we have the following decomposition:

$$\ell_{\text{kd}} = \underbrace{-\mathbf{p}_y^T(\tau)\log\mathbf{p}_y^S(\tau)}_{\text{Correct Guidance}} \underbrace{-\sum_{c\neq y} e\left(\mathbf{q}^T(\tau)\right)\log\mathbf{p}_c^S(\tau)}_{\text{Smooth Regularization}} \underbrace{-\sum_{c\neq y}\left(\mathbf{p}_c^T(\tau) - e\left(\mathbf{q}^T(\tau)\right)\right)\log\mathbf{p}_c^S(\tau)}_{\text{Class Discriminability}}. \quad (2)$$

(I) Correct Guidance: this term guarantees correctness during teaching. The decomposition in [9] also contains this term, which is explained as importance weighting. This term works similarly to the cross-entropy loss, which could be dealt with separately when applying temperature scaling.

(II) Smooth Regularization: some previous works [55, 62, 63] attribute the success of KD to the efficacy of regularization and study its relation to label smoothing (LS). The combination of this term with *correct guidance* works similarly to LS. Notably, $e\left(\mathbf{q}^T(\tau)\right)$ differs across samples, implying that the strength of smoothing is instance-specific, which is similar to the analysis in [62].

(III) Class Discriminability: this term tells the student the affinity of wrong classes to the correct class. Transferring the knowledge of class similarities to students has been the mainstream guess of the "dark knowledge" in KD [13, 38]. Ideally, a good teacher should be as discriminating as possible in telling students which classes are more related to the correct class.

Illustrations of the decomposition are presented in the left of Fig. 1. Obviously, an appropriate temperature should simultaneously contain the efficacy of the three terms, e.g., the shaded row in Fig. 1. A too high or too low temperature could lead to smaller *class discriminability*, making the guidance less different among wrong classes, which weakens the distillation performance in practical. Among these three terms, we advocate that *class discriminability* is more fundamental in KD and present more discussions in Appendix A (verified in Fig. 2 and Fig. 3).

To measure these three terms quantitatively, we use the *target class probability* (i.e., $\mathbf{p}_y$), the *average of wrong class probabilities* (i.e., $e(\mathbf{q}) = \frac{1}{C-1}\sum_{j\neq y}\mathbf{p}_j$), and the *variance of wrong class probabilities* (i.e., $v(\mathbf{q}) = \frac{1}{C-1}\sum_{j\neq y}\left(\mathbf{p}_j - e(\mathbf{q})\right)^2$) as estimators. $e(\cdot)$ and $v(\cdot)$ calculates the mean and variance of the elements in a vector. In some cases, we use the standard deviation as an estimator for the third term, i.e., $\sigma(\mathbf{q}) = v^{1/2}(\mathbf{q})$. Because the latter two terms are calculated after applying softmax to the complete logit vector, we define them as *Derived Average (DA)* and *Derived Variance (DV)*, respectively. In experiments, we calculate these metrics for all training samples and sometimes report the average or standard deviation across these samples.

4.2 Theoretical Analysis

We analyze the mean and variance of the softened probability vector, i.e., the teacher's label $\mathbf{p}^T(\tau)$ used in KD. We defer the proofs of Lemma 4.1 and Proposition 4.3, 4.4 to Appendix B.

Lemma 4.1 (Variance of Softened Probabilities)**.** *Given a logit vector* $\mathbf{f} \in \mathbb{R}^C$ *and the softened probability vector* $\mathbf{p} = SF(\mathbf{f};\tau), \tau \in (0,\infty)$, $v(\mathbf{p})$ *monotonically decreases as* τ *increases.*

As τ increases, $\mathbf{p}(\tau)$ becomes more uniform, i.e., it's entropy increases. However, we especially focus on the wrong classes, where the mean and variance are more intuitive to calculate and analyze.

Assumption 4.2. The target logit is higher than other classes' logits, i.e., $\mathbf{f}_y \geq \mathbf{f}_c, \forall c \neq y$.

Assumption 4.2 is rational because well-performed teachers could almost achieve a higher accuracy (e.g., >95%) on the training set, and most training samples meet this requirement.

Proposition 4.3. *Under Assumption 4.2,* $\mathbf{p}_y$ *monotonically decreases as* τ *increases, and* $e(\mathbf{q})$ *monotonically increases as* τ *increases. As* $\tau \to \infty$, $e(\mathbf{q}) \to 1/C$.

Proposition 4.3 implies that increasing temperature could lead to a higher *derived average* (empirically see Fig. 7) and strengthen the *smooth regularization* term in Eq. 2.

Before we analyze the *class discriminability* term, we define $\tilde{\mathbf{q}}(\tau)$ as the result of applying softmax *only to the wrong logits* with temperature τ, i.e., $\tilde{\mathbf{q}}(\tau) = \text{SF}(\mathbf{g};\tau)$. For the element index c' of $\mathbf{q}$, we have

$$\tilde{\mathbf{q}}_{c'}(\tau) = \exp(\mathbf{g}_{c'}/\tau) / \sum_j \exp(\mathbf{g}_j/\tau). \tag{3}$$

Notably, $\tilde{\mathbf{q}}$ differs from $\mathbf{q}$ a lot. Specifically, the former satisfies $\sum_{c'} \tilde{\mathbf{q}}_{c'} = 1$, while the summation of the latter is $\sum_{c \neq y} \mathbf{p}_c = 1 - \mathbf{p}_y$. The former does not depend on the correct class's logit while the latter does. We name $v(\tilde{\mathbf{q}})$ *Inherent Variance (IV)* because it only depends on wrong classes' logits.

Proposition 4.4 (Derived Variance vs. Inherent Variance)**.** *The derived variance is determined by the square of derived average and the inherent variance via:*

$$\underbrace{v(\mathbf{q})}_{DV} = (C-1)^2 \underbrace{e^2(\mathbf{q})}_{DA^2} \underbrace{v(\tilde{\mathbf{q}})}_{IV}. \tag{4}$$

With τ increases, $e(\mathbf{q})$ increases (Proposition 4.3) while $v(\tilde{\mathbf{q}})$ decreases (Lemma 4.1), and hence, it is not so easy to judge the specific monotonicity of $v(\mathbf{q})$ w.r.t. τ. Empirically, we observe that the *derived variance* first increases and then decreases (see Fig. 7), which conforms to the change of the *class discriminability* as illustrated in Fig. 1.

We could use Proposition 4.4 to clearly analyze why larger teacher networks can't teach well. Before this, we present another two properties and a corollary without detailed proof.

Remark 4.5. Fixing $\mathbf{g}$ and τ, a higher target logit $\mathbf{f}_y$ leads to a higher $\mathbf{p}_y$, i.e., a smaller *derived average* $e(\mathbf{q})$.

Remark 4.6. Fixing τ, less varied wrong logits $\mathbf{g}$ leads to less varied $\tilde{\mathbf{q}}$, i.e., a smaller *inherent variance* $v(\tilde{\mathbf{q}})$.

Corollary 4.7. *Suppose we have two teachers T_1 and T_2, and their logit vectors for a same sample are $\mathbf{f}^{T_1}$ and $\mathbf{f}^{T_2}$.*

- *If $\mathbf{f}_y^{T_1} \geq \mathbf{f}_y^{T_2}$ while $\mathbf{g}^{T_1}$ and $\mathbf{g}^{T_2}$ are nearly the same, then $\mathbf{p}_y^{T_1} \geq \mathbf{p}_y^{T_2}$ (Remark 4.5) while $v(\tilde{\mathbf{q}}^{T_1}) \approx v(\tilde{\mathbf{q}}^{T_2})$. Hence, $v(\mathbf{q}^{T_1}) \leq v(\mathbf{q}^{T_2})$.*
- *If $\mathbf{f}_y^{T_1} \approx \mathbf{f}_y^{T_2}$ while $v(\mathbf{g}^{T_1}) \leq v(\mathbf{g}^{T_2})$, then $\mathbf{p}_y^{T_1} \approx \mathbf{p}_y^{T_2}$ while $v(\tilde{\mathbf{q}}^{T_1}) \leq v(\tilde{\mathbf{q}}^{T_2})$ (Remark 4.6). Hence, $v(\mathbf{q}^{T_1}) \leq v(\mathbf{q}^{T_2})$.*

This corollary explains why a larger teacher can't teach better. Because the larger teacher tends to be over-confident, the target logit $\mathbf{f}_y$ may be larger or the variance of wrong logits $v(\mathbf{g})$ may be smaller. These are illustrated in Fig. 1 and empirically verified in Fig. 4. Then the *derived variance* $v(\mathbf{q})$ may be smaller, limiting the efficacy of *class discriminability* in Eq. 2. Empirical results are in Fig. 7.

Notably, we focus on analyzing the variance of wrong class probabilities instead of all classes. Maximizing the variance of all classes' probabilities does not mean maximizing the variance of wrong classes'. For example, although a very low temperature can maximize the variance of all classes' probabilities, the generated teacher's label is one-hot that shows no distinctness between wrong classes. In other words, the effectiveness of KD should be more related to the distinctness between wrong classes rather than all classes. However, traditional temperature scaling applies a uniform temperature for all classes, which cannot separately handle the wrong classes.

4.3 Asymmetric Temperature Scaling

We conclude the above analysis: *if a larger teacher makes an over-confident prediction, the wrong class probabilities it provides could be not discriminative enough.* Utilizing a uniform temperature could not enlarge the *derived variance* as much as possible with the interference of the target class's logit (see the middle of Fig. 7). Thanks to the decomposition in Eq. 2, the *correct guidance* term works similarly to the cross-entropy loss and allows us to deal with it separately. Hence, we propose a novel temperature scaling approach:

$$\mathbf{p}_c(\tau_1, \tau_2) = \exp\left(\mathbf{f}_c/\tau_c\right) / \sum_{j \in [C]} \exp\left(\mathbf{f}_j/\tau_j\right), \qquad \tau_i = \mathcal{I}\{i = y\}\tau_1 + \mathcal{I}\{i \neq y\}\tau_2, \forall i \in [C], \tag{5}$$

where we take $\tau_1 > \tau_2 > 0$. This approach is named *Asymmetric Temperature Scaling (ATS)* because we apply different temperatures to the logits of correct and wrong classes. According to Eq. 4, ATS could bring such benefits when the teacher is over-confident:

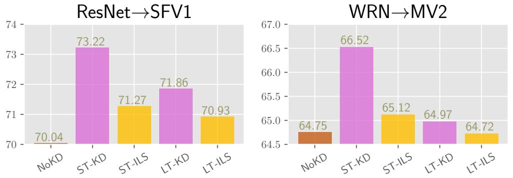

Figure 2: Student's test accuracies without KD ("NoKD"), with KD ("-KD"), and only with the first two terms in Eq. 2 ("-ILS"). "ST"/"LT" refers to "small/large teacher".

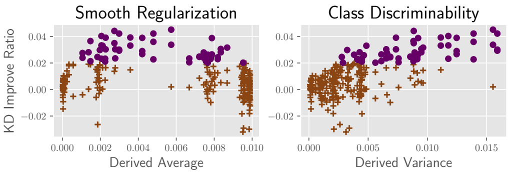

Figure 3: Correlations of *smooth regularization* (measured by *derived average*) and *class discriminability* (measured by *derived variance*) w.r.t. KD improvement ratio.

- If the teacher outputs a larger logit $\mathbf{f}_y$ for the correct class, a relatively larger τ_1 could decrease it to a reasonable magnitude, i.e., decreasing $\mathbf{p}_y$ and increasing $e(\mathbf{q})$, and finally increasing the *derived variance* $v(\mathbf{q})$;
- If the teacher outputs less varied logits $\mathbf{g}$ for wrong classes, a relatively smaller temperature τ_2 could make them more diverse, i.e., increasing $v(\tilde{\mathbf{q}})$, finally increasing the *derived variance* $v(\mathbf{q})$.

ATS is more flexible in enlarging the *derived variance* (see the right of Fig. 7), i.e., it could generate more discriminative distillation guidance during teaching. Take the demo in Fig. 1 as an example, the smaller and larger teacher captures the same relative class affinities, e.g., they both know that the fourth/fifth class (shaded cells) is the most/least relevant to the target class. However, with a uniform temperature 4.0, the smaller teacher provides probabilities $(0.073, 0.060)$ for these two classes, while over-confident larger teachers provide $(0.040, 0.033)$ or $(0.074, 0.067)$. Clearly, the absolute affinities of the larger teachers are not so discriminative as the smaller teacher's. Utilizing ATS, we could respectively apply $(\tau_1 = 4.67, \tau_2 = 4.0)$ or $(\tau_1 = 4.0, \tau_2 = 2.0)$ to the over-confident teacher's logits, generating the same probability vector as the smaller teacher's. ATS utilizes two temperatures and creates the wiggle room to make the distribution over wrong classes more discriminative.

5 Experiments

We use CIFAR-10/CIFAR-100 [21], TinyImageNet [43], CUB [47], Stanford Dogs [19], and Google Speech Commands [48] as the datasets. For teacher networks, we use different versions of ResNet [11], WideResNet [56], ResNeXt [49]. For student networks, we use VGG [39], ShuffleNetV1/V2 [59, 29], AlexNet [22], MobileNetV2 [37], and DSCNN [60].

We majorly follow the training settings in [44] [1]. Except that the Google Speech Commands takes 50 epochs, we train networks on other datasets with 240 epochs. We use the SGD optimizer with 0.9 momentum. For VGG, AlexNet, ResNet, WideResNet, and ResNeXt, we set the learning rate as 0.05 (recommended by [44]). For ShuffleNet and MobileNet, we use a smaller learning rate of 0.01 (recommended by [44]). We use the pre-trained models provided in PyTorch for CUB and Stanford Dogs, and correspondingly, their learning rates are scaled by $0.1\times$. During training, we decay the learning rate by 0.1 every 30 epochs after the first 150 epochs (recommended by [44]). For Google Speech Commands, we decay the learning rate via the cosine annealing. We set the batch size as 128 for CIFAR data, 64 for other datasets. Other dataset, network and training details are in Appendix C.

5.1 Observations

Class discriminability matters a lot in KD and correlates with the KD improvement. To show the importance of *class discriminability* in KD, we omit the distinctness of wrong class probabilities during distillation. Specifically, we only keep the first two terms in Eq. 2, which works similarly to the instance-specific label smoothing (abbreviated as "ILS"). Fig. 2 shows students' performances on CIFAR-100. Without the third term in Eq. 2, both small teachers ("ST") and larger teachers ("LT") teach worse significantly. Then, we investigate the correlations of KD performance improvement (i.e., the test accuracy change ratio with KD w.r.t. without KD) with *smooth regularization* and *class*

[1] `https://github.com/HobbitLong/RepDistiller`

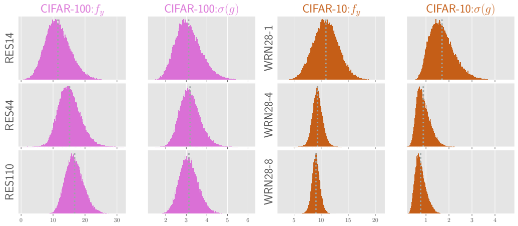

Figure 4: The distributions of the *target logit* ($\mathbf{f}_y$) and the *standard deviation of wrong logits* ($\sigma(\mathbf{g})$) of the 50K training samples on CIFAR-10/100. Rows show networks with various capacity.

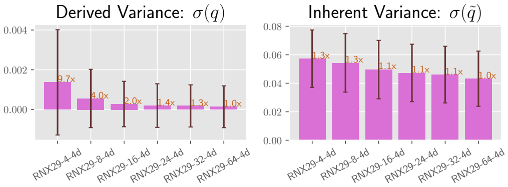

Figure 5: The *derived variance* and *inherent variance* on CIFAR-100 using ResNeXt. Each bar shows the mean and standard deviation across 50K training samples. The x-axis shows networks with various capacity.

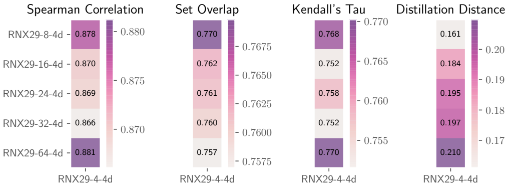

Figure 6: Several metrics among teachers with various capacities on CIFAR-10. The first three metrics are related to the relative magnitudes of teachers' label, while distillation distance is related to the absolute magnitudes.

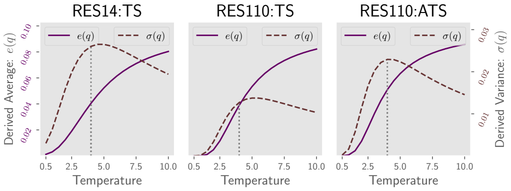

Figure 7: The change of *derived average* ($e(\mathbf{q})$) and *derived variance* ($v(\mathbf{q})$) as τ increases from 0.1 to 10.0 on CIFAR-10. The third one shows the results of ResNet110 with the proposed ATS. *DV* under TS is limited while ATS enlarges it.

discriminability under 270 pairs of "(teacher, student, temperature)". Details are in Appendix C.4. Fig. 3 plots the scatters, where dots show pairs whose improvement is higher than 2%. Clearly, teachers with a larger *derived variance* tend to guide better. *These observations show the rationality of the proposed KD decomposition and imply that enhancing derived variance is beneficial*.

Larger teachers provide a larger target logit or less varied wrong logits. We first compare the logit distributions provided by larger and smaller teachers on the training set. Fig. 4 plots the histograms (200 bins) of the target logit (i.e., $\mathbf{f}_y$) and the standard deviation of wrong logits (i.e., $\sigma(\mathbf{g})$). The left and right, respectively, show the results on CIFAR-100 and CIFAR-10. Clearly, the first column shows that ResNet110 tends to generate a larger target logit (i.e., $\mathbb{E}_{\mathbf{x}}[\mathbf{f}_y] \approx 15.0$) than ResNet14 (i.e., $\mathbb{E}_{\mathbf{x}}[\mathbf{f}_y] \approx 10.0$). On CIFAR-10, the smallest $\mathbf{f}_y$ given by WRN28-8 is larger than WRN28-1. Furthermore, WRN28-8 gives smaller variance (the fourth column), i.e., smaller *inherent variance*. *These reveal specific manifestations of complex networks' over-confidence*.

Larger and smaller teachers have similar inherent variance while different derived variance under traditional temperature scaling. We use $\tau = 1.0$ to soften the *complete logits* and *only the wrong class logits*, respectively, and then show the mean and standard deviation of *derived variance* and *inherent variance* across training samples in Fig. 5. Although the larger models' *inherent variance* is smaller, the difference between RNX29-4-4d and RNX29-64-4d is only up to $1.3\times$. However, the *derived variance* differs a lot, where the smaller teacher's variance is approximately $9.7\times$ as the larger teacher's. These observations imply that *traditional temperature scaling could seriously decrease the derived variance of larger teachers though they have appreciable inherent variance*.

Complex teachers know approximately the same as smaller teachers on relative class affinities. Only if the relative magnitudes of the wrong class probabilities are correct, it is valid to enlarge the discrimination between them. Otherwise, if a teacher himself misunderstands the knowledge's principles, it will be counterproductive to reinforce this knowledge to students. Given two teachers T_1 and T_2, we first calculate the *Spearman correlation* between $\mathbf{p}^{T_1}$ and $\mathbf{p}^{T_2}$. Second, the set overlap between the top-5 predictions is calculated, i.e., $|\mathcal{C}^{T_1} \cap \mathcal{C}^{T_2}|/|\mathcal{C}^{T_1} \cup \mathcal{C}^{T_2}|$. $\mathcal{C}$ denotes the set of top-5 predicted classes. Third, we calculate Kendall's τ between $\mathbf{p}^{T_1}$ and $\mathbf{p}^{T_2}$, which directly shows the rank correlation of two teachers. These metrics only depend on classes' relative magnitudes. The results are in Fig. 6. Excitingly, these metrics among teachers with different capacities do not vary a

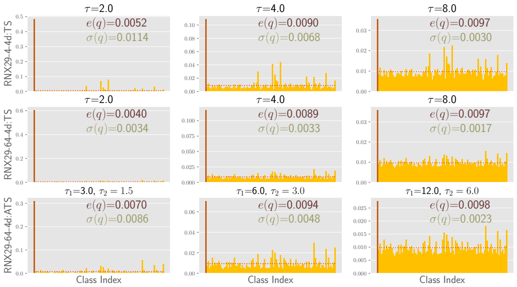

Figure 8: Probability vector visualization of a randomly selected training sample from CIFAR-100. The target class is $y = 1$. The bottom row shows applying ATS to the larger teacher.

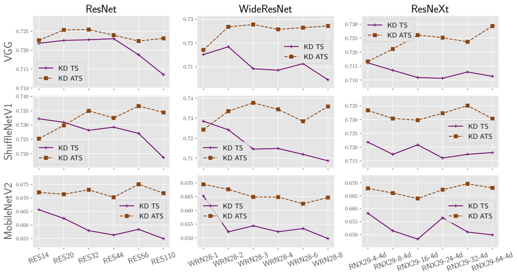

Figure 9: Distillation results via TS (solid curves) and ATS (dashed curves) on CIFAR-100. The x-axis of each figure shows teachers with various capacities.

Table 2: Comparisons with SOTA methods on CIFAR-100. ResNet110, WRN28-8, and RNX29-64-4d are teachers. VGG8, SFV1, and MV2 are students. The area in gray shows the results of the ensemble. "KD+ATS" and "KD+ATS+Ens" are our methods.

Teacher	ResNet110 (74.09)			WRN28-8 (79.73)			RNX29-64-4d (79.91)			Avg
Student	VGG8	SFV1	MV2	VGG8	SFV1	MV2	VGG8	SFV1	MV2	
NoKD	69.92	70.04	64.75	69.92	70.04	64.75	69.92	70.04	64.75	68.24
ST-KD	72.30	73.22	66.56	71.85	72.85	**66.52**	71.61	72.18	65.82	70.32
KD	71.35	71.86	65.49	70.46	70.87	64.97	71.13	71.80	64.99	69.21
ESKD	71.88	72.02	65.92	71.13	71.32	65.09	71.09	71.27	64.83	69.39
TAKD	**72.71**	72.86	66.98	71.20	71.62	65.11	71.46	71.44	65.36	69.86
SCKD	70.38	70.61	64.59	70.83	70.52	65.19	70.33	70.92	64.86	68.69
KD+ATS	72.31	**73.44**	**67.18**	**72.72**	**73.58**	66.47	**72.93**	**73.03**	**66.80**	**70.94**
Ens	72.77	73.61	67.76	72.77	73.61	67.76	72.77	73.61	67.76	71.38
ResKD	73.89	**76.03**	69.00	73.84	**75.14**	67.69	74.64	75.43	68.10	72.64
KD+ATS+Ens	**74.86**	75.05	**69.50**	**74.60**	75.04	**68.79**	**75.34**	**75.47**	**69.82**	**73.16**

lot, and the Spearman correlations are almost all larger than 0.85. According to the interpretation of Kendall's τ [8, 53], if the smaller teacher predicts that class i is more related to the target class than that of class j, then the larger teacher has a probability of $(0.75 + 1)/2 = 0.875$ to give the same relative affinity. As a comparison, we also calculate the distillation distance defined in [15], which utilizes L1 distance and depends on the absolute magnitudes of probabilities. Using this metric, the distance increases quickly as the capacity gap increases. These observations demonstrate that *teachers know approximately the same about relative class affinities while their absolute values differ significantly under traditional temperature scaling*.

The proposed ATS could enlarge the derived variance of larger teachers. The above analysis verifies that the over-confident teachers experience lower *derived variance* under traditional temperature scaling although they grasp the relative class affinities well. For an intuitive visualization, we plot the probabilities for a randomly sampled instance whose correct class is $y = 1$ from CIFAR-100. The top two rows in Fig. 8 show the results of RNX29-4-4d and RNX29-64-4d under traditional temperature scaling (TS), where the latter really experiences a smaller *derived variance*. Using ATS could enhance the *derived variance* as shown at the bottom row (the bars are more jagged). Then, we study the change of *derived average* (DA) and *derived variance* (DV) as temperature increases. Given a τ, we obtain the DA and DV for all training samples via softmax and then calculate the average. For ATS, we use $\tau_1 = 1.25\tau$ and $\tau_2 = 0.75\tau$. The curves are shown in Fig. 7. According to Proposition 4.3, $e(\mathbf{q})$ increases as τ increases, which enhances the efficacy of *smooth regularization*. Notably, this term changes nearly the same between teachers with various capacities. However, the *derived variance* $v(\mathbf{q})$ differs a lot. Empirically, $v(\mathbf{q})$ first increases and then decreases, and the maximal of the larger teacher's DV is smaller, which verifies the Corollary 4.7. Because *derived variance* corresponds to the efficacy of *class discriminability*, this shows why larger teachers can't teach well. Using the proposed ATS could enhance the *derived variance*, which equivalently improves

Table 3: Comparisons with SOTA methods on TinyImageNet, CUB, and Stanford Dogs. WRN50-2 and RNX101-32-8d are teachers. AlexNet, SFV2, and MV2 are students.

	TinyImageNet			CUB			Stanford Dogs			Avg
Teacher	WRN50-2 (66.28)			RNX101-32-8d (79.50)			RNX101-32-8d (73.98)			
Student	ANet	SFV2	MV2	ANet	SFV2	MV2	ANet	SFV2	MV2	
NoKD	34.62	45.79	52.03	55.66	71.24	74.49	50.20	68.72	68.67	57.94
ST-KD	36.16	49.59	52.93	56.39	72.15	76.80	51.95	69.92	72.06	59.77
KD	35.83	48.48	52.33	55.10	71.89	76.45	50.22	68.48	71.25	58.89
ESKD	34.97	48.34	52.15	55.64	72.15	76.87	50.39	69.02	71.56	59.01
TAKD	36.20	48.71	52.44	54.82	71.53	76.25	50.36	68.94	70.61	58.87
SCKD	36.16	48.76	51.83	56.78	71.99	75.13	51.78	68.80	70.13	59.04
KD+ATS	**37.42**	**50.03**	**54.11**	**58.32**	**73.15**	**77.83**	**52.96**	**70.92**	**73.16**	**60.88**
Ens	39.37	50.69	56.40	59.84	74.43	77.47	54.04	71.65	72.53	61.82
ResKD	38.66	51.93	57.32	**62.60**	75.29	76.27	54.68	70.73	72.85	62.26
KD+ATS+Ens	**40.42**	**52.14**	**58.47**	62.00	**76.26**	**78.97**	**55.69**	**73.22**	**74.67**	**63.54**

the efficacy of *class discriminability* in KD. We conclude that *traditional temperature scaling leads to distillation labels with less discriminative information among wrong class probabilities; our proposed ATS could enhance the discrimination among wrong classes and benefit the distillation process.*

5.2 Performances

ATS makes larger teachers teach well again. Previous studies find that more accurate teachers can't necessarily teach well [6, 31]. As shown in Fig. 9, although we tune temperatures in $\{1.0, 2.0, 4.0, 8.0, 12.0, 16.0\}$, larger teachers still teach worse under traditional temperature scaling (the solid curves). However, using ATS (the dashed curves) could make larger teachers teach well or better again. The details are in Appendix C.4.

ATS surpasses previous methods with advanced techniques. We compare with SOTA methods and list the results on CIFAR-100, TinyImageNet, CUB, and Dogs in Tab. 2 and Tab. 3. The sota compared methods include ESKD [6], TAKD [31], SCKD [51], and ResKD [10, 24]. NoKD trains students without the teacher's supervision. ST-KD trains students under the guidance of a smaller teacher. KD trains students under the guidance of the larger teacher. More details of compared methods are in Appendix C.3. The last column of these tables shows the average performance of corresponding rows. The larger teacher could slightly improve the students' performances via traditional KD, i.e., $68.24\% \rightarrow 69.21\%$ and $57.94 \rightarrow 58.89\%$. However, using a smaller teacher (the row of "ST-KD") could obtain about 2% improvement on average, i.e., $68.24\% \rightarrow 70.32\%$ and $57.94\% \rightarrow 59.77\%$. This again verifies that larger teachers really teach worse on various datasets. Taking advantage of the *early-stopped teacher* (ESKD), the *teacher assistant* (TAKD), and the *student customized teacher* (SCKD) only improves the larger teacher slightly, e.g., $69.21\% \rightarrow 69.86\%$ and $58.89\% \rightarrow 59.04\%$. These results do not even surpass the small teacher. ResKD improves the students' performances a lot via introducing the *residual student* and taking the two students' ensemble, which surpasses the ensemble performances of two separately trained students under different initializations (the "Ens" row). For a fair comparison, we also test the performances of our methods via repeating the "KD+ATS" two times and making predictions via the ensemble. The results are almost 1% higher than ResKD. Results on CIFAR-10 and speech data are in Appendix D.

Ablation Studies One might argue that the validity of ATS is due to the tuning from a larger hyper-parameter space. We vary τ_1 from 1.0 to 6.0, τ_2 from 1.0 to 5.0, and record the performances in Fig. 10. Obviously, setting $\tau_1 > \tau_2$ could be better, and especially, we recommend the setting of $\tau_2 \in [\tau_1 - 2, \tau_1 - 1]$. Although we introduce one more hyper-parameter, ATS is simple to implement and surpasses SOTA methods that take advanced techniques. We achieve the goal of only utilizing simple operations to make larger teachers teach well again. Then, we study the trade off of the KD loss and CE loss under different λ. The results of "WRN28-8 → MV2" on CIFAR-100 (Fig. 11) verifies that ATS could improve the performances under various λ. We also compare ATS with other types of KD under various scenes, and the results are in Appendix D.3.

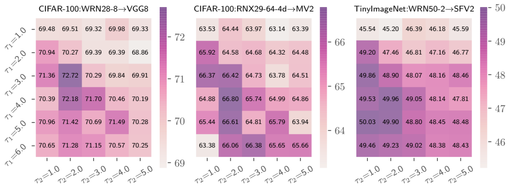

Figure 10: Ablation studies on asymmetric temperatures on CIFAR-100 and TinyImageNet (τ_1, τ_2 in Eq. 5).

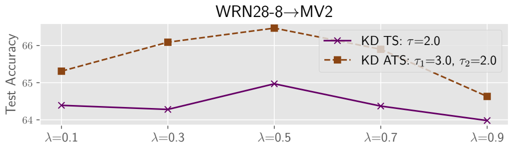

Figure 11: Ablation studies on the weighting of KD loss and CE loss on CIFAR-100 (λ in Eq. 1).

6 Conclusion

We study the miraculous phenomenon in KD that a more accurate model doesn't necessarily teach better. The proposed KD decomposition attributes the success of a better teacher to three factors, including *correct guidance*, *smooth regularization*, and *class discriminability*. Through theoretical analysis, over-confident teachers could not release their potential abilities of the *class discriminability* under traditional temperature scaling. As a simple yet effective solution, we propose *Asymmetric Temperature Scaling (ATS)* to enhance the *derived variance* of larger teachers, making their distillation labels more discriminative when teaching students. Extensive experimental results verify the superiorities of our proposed methods.

7 Broader Impact

We focus on the variance of wrong class probabilities to analyze why larger teachers cannot teach well and hope that our research could bring a new perspective to the KD field. Our work has no potential negative societal impacts.

Acknowledgements

This work is partially supported by National Natural Science Foundation of China (Grant No. 61921006, 41901270), and Natural Science Foundation of Jiangsu Province (Grant No. BK20190296). Thanks to Huawei Noah's Ark Lab NetMIND Research Team. Professor De-Chuan Zhan is the corresponding author.

References

[1] Sungsoo Ahn, Shell Xu Hu, Andreas C. Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019.

[2] Rohan Anil, Gabriel Pereyra, Alexandre Passos, Róbert Ormándi, George E. Dahl, and Geoffrey E. Hinton. Large scale distributed neural network training through online distillation. In *The 6th International Conference on Learning Representations*, 2018.

[3] Cristian Bucila, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–541, 2006.

[4] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5008–5017, 2021.

[5] Xu Cheng, Zhefan Rao, Yilan Chen, and Quanshi Zhang. Explaining knowledge distillation by quantifying the knowledge. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12922–12932, 2020.

[6] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *IEEE/CVF International Conference on Computer Vision*, pages 4793–4801, 2019.

[7] Tri Dao, Govinda M. Kamath, Vasilis Syrgkanis, and Lester Mackey. Knowledge distillation as semiparametric inference. In *The 9th International Conference on Learning Representations*, 2021.

[8] Ronald Fagin, Ravi Kumar, and D. Sivakumar. Comparing top k lists. *SIAM Journal on Discrete Mathematics*, 17(1):134–160, 2003.

[9] Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 1602–1611, 2018.

[10] Mengya Gao, Yujun Wang, and Liang Wan. Residual error based knowledge distillation. *Neurocomputing*, 433:154–161, 2021.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *The Thirty-Third AAAI Conference on Artificial Intelligence*, pages 3779–3787, 2019.

[13] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[14] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[15] Daniel Hsu, Ziwei Ji, Matus Telgarsky, and Lan Wang. Generalization bounds via distillation. In *The 9th International Conference on Learning Representations*, 2021.

[16] Zhen Huang, Xu Shen, Jun Xing, Tongliang Liu, Xinmei Tian, Houqiang Li, Bing Deng, Jianqiang Huang, and Xian-Sheng Hua. Revisiting knowledge distillation: An inheritance and exploration framework. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3579–3588, 2021.

[17] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *CoRR*, abs/1707.01219, 2017.

[18] Guangda Ji and Zhanxing Zhu. Knowledge distillation in wide neural networks: Risk bound, data efficiency and imperfect teacher. In *Advances in Neural Information Processing Systems 33*, 2020.

[19] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei fei Li. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Categorization, CVPR (2011).*

[20] Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. Self-knowledge distillation with progressive refinement of targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6567–6576, 2021.

[21] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2012.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1106–1114, 2012.

[23] Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *Proceedings of the 35th International Conference on Machine Learning*, pages 2830–2839, 2018.

[24] Xuewei Li, Songyuan Li, Bourahla Omar, Fei Wu, and Xi Li. Reskd: Residual-guided knowledge distillation. *IEEE Transactions on Image Processing*, 30:4735–4746, 2021.

[25] Ruofan Liang, Tianlin Li, Longfei Li, Jing Wang, and Quanshi Zhang. Knowledge consistency between neural networks and beyond. In *The 8th International Conference on Learning Representations*, 2020.

[26] Junjie Liu, Dongchao Wen, Hongxing Gao, Wei Tao, Tse-Wei Chen, Kinya Osa, and Masami Kato. Knowledge representing: Efficient, sparse representation of prior knowledge for knowledge distillation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 638–646, 2019.

[27] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 7096–7104, 2019.

[28] David Lopez-Paz, Léon Bottou, Bernhard Schölkopf, and Vladimir Vapnik. Unifying distillation and privileged information. In *The 4th International Conference on Learning Representations*, 2016.

[29] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: practical guidelines for efficient CNN architecture design. In *Computer Vision - ECCV 2018 - 15th European Conference*, pages 122–138, 2018.

[30] Aditya Krishna Menon, Ankit Singh Rawat, Sashank J. Reddi, Seungyeon Kim, and Sanjiv Kumar. A statistical perspective on distillation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 7632–7642, 2021.

[31] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 5191–5198, 2020.

[32] Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems 32*, pages 4696–4705, 2019.

[33] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Computer Vision - ECCV 2018 - 15th European Conference*, volume 11215, pages 283–299, 2018.

[34] Baoyun Peng, Xiao Jin, Dongsheng Li, Shunfeng Zhou, Yichao Wu, Jiaheng Liu, Zhaoning Zhang, and Yu Liu. Correlation congruence for knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 5006–5015, 2019.

[35] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *Proceedings of the 36th International Conference on Machine Learning*, pages 5142–5151, 2019.

[36] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *The 3rd International Conference on Learning Representations*, 2015.

[37] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[38] Zhiqiang Shen, Zechun Liu, Dejia Xu, Zitian Chen, Kwang-Ting Cheng, and Marios Savvides. Is label smoothing truly incompatible with knowledge distillation: An empirical study. In *The 9th International Conference on Learning Representations*, 2021.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations*, 2015.

[40] Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H. Chi, and Sagar Jain. Understanding and improving knowledge distillation. *CoRR*, abs/2002.03532, 2020.

[41] Raphael Tang and Jimmy Lin. Deep residual learning for small-footprint keyword spotting. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5484–5488, 2018.

[42] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *The 5th International Conference on Learning Representations*, 2017.

[43] Amirhossein Tavanaei. Embedded encoder-decoder in convolutional networks towards explainable AI. *CoRR*, abs/2007.06712, 2020.

[44] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *The 8th International Conference on Learning Representations*, 2020.

[45] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 1365–1374, 2019.

[46] Gregor Urban, Krzysztof J. Geras, Samira Ebrahimi Kahou, Özlem Aslan, Shengjie Wang, Abdelrahman Mohamed, Matthai Philipose, Matthew Richardson, and Rich Caruana. Do deep convolutional nets really need to be deep and convolutional? In *The 5th International Conference on Learning Representations*, 2017.

[47] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

[48] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *CoRR*, abs/1804.03209, 2018.

[49] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5987–5995, 2017.

[50] Han-Jia Ye, Su Lu, and De-Chuan Zhan. Distilling cross-task knowledge via relationship matching. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12393–12402, 2020.

[51] Yi Wang Yichen Zhu. Student customized knowledge distillation: Bridging the gap between student and teacher. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[52] Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 7130–7138, 2017.

[53] Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. Logme: Practical assessment of pre-trained models for transfer learning. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12133–12143, 2021.

[54] Lu Yu, Vacit Oguz Yazici, Xialei Liu, Joost van de Weijer, Yongmei Cheng, and Arnau Ramisa. Learning metrics from teachers: Compact networks for image embedding. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2907–2916, 2019.

[55] Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3902–3910, 2020.

[56] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016.

[57] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *The 5th International Conference on Learning Representations*, 2017.

[58] Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *2019 IEEE/CVF International Conference on Computer Vision*, pages 3712–3721, 2019.

[59] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

[60] Yundong Zhang, Naveen Suda, Liangzhen Lai, and Vikas Chandra. Hello edge: Keyword spotting on microcontrollers. *CoRR*, abs/1711.07128, 2017.

[61] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. Deep mutual learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.

[62] Zhilu Zhang and Mert R. Sabuncu. Self-distillation as instance-specific label smoothing. In *Advances in Neural Information Processing Systems 33*, 2020.

[63] Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. In *The 9th International Conference on Learning Representations*, 2021.

Checklist

1. For all authors...
    (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
    (b) Did you describe the limitations of your work? [Yes] Introducing one more hyperparameter.
    (c) Did you discuss any potential negative societal impacts of your work? [Yes] No potential negative societal impacts.
    (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
    (a) Did you state the full set of assumptions of all theoretical results? [Yes]
    (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
    (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No]
    (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
    (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
    (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
    (a) If your work uses existing assets, did you cite the creators? [N/A]
    (b) Did you mention the license of the assets? [N/A]
    (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
    (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
    (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
    (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
    (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
    (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A More Discussions About KD Decomposition (Eq. 2)

We decompose KD into three terms as shown in Eq. 2, i.e., the *correct guidance*, the *smooth regularization*, and the *class discriminability*. We take a reverse thinking that *how can we construct these three terms if without KD?* First, to ensure the learning process of the student is correct, the target class's label should be the highest. This is simple because we have one-hot labels of training samples. That is, the cross-entropy loss could guarantee the student learns correctly. Second, how can we introduce *smooth regularization*? This is also simple because we could utilize label smoothing (LS) as regularizations. However, using cross-entropy and LS is still not as effective as KD. Hence, the *class discriminability* also matters a lot. This term is not easy to construct because we do not have the prior class similarities or the instance-level class similarities. Fortunately, a teacher may contain the information of class similarities in KD. We could find that the *class discriminability* may matter more through the above analysis. Hence, we focus on its efficacy when analyzing why larger models can't teach well.

Furthermore, the *correct guidance* is closely related to the second term, i.e., *smooth regularization*, as $e(\mathbf{q}) = \frac{1}{C-1}(1 - \mathbf{p}_y)$. That is, the strength of *correct guidance* is negatively correlated with the strength of *smooth regularization*. As shown in Fig. 7, after applying ATS, the smooth regularization curves in these three subfigures are nearly the same, which indicates that the *correct guidance* term is slightly affected when utilizing ATS.

B Detailed Proofs of Lemma 4.1, Proposition 4.3 and Proposition 4.4

We first present some basic results about softmax operations. The softmax operation is $\mathbf{p}_c = \frac{\exp(\mathbf{f}_c/\tau)}{Z}$, where $Z = \sum_{j=1}^{C} \exp(\mathbf{f_j}/\tau)$. We then present some basic derivatives:

$$\frac{\partial Z}{\partial \tau} = -\sum_{c=1}^{C} \frac{\mathbf{f}_c}{\tau^2} \exp(\mathbf{f}_c/\tau), \quad \frac{\partial \mathbf{p}_c}{\partial \tau} = \frac{\mathbf{p}_c}{\tau^2} \left(\sum_{j=1}^{C} \mathbf{p}_j \mathbf{f}_j - \mathbf{f}_c \right) \tag{6}$$

Lemma B.1 (Variance of Softened Probabilities (Lemma 4.1 in the Body))**.** *Given a logit vector $\mathbf{f} \in \mathbb{R}^C$ and the softened probability vector via softmax function, i.e., $\mathbf{p} = SF(\mathbf{f}; \tau), \tau \in (0, \infty)$, $v(\mathbf{p})$ monotonically decreases as τ increases.*

Proof. Obviously, $e(\mathbf{p}) = \frac{1}{C}$. Then, we have:

$$v(\mathbf{p}) = \frac{1}{C} \sum_{c=1}^{C} (\mathbf{p}_c - e(\mathbf{p}))^2 = \frac{1}{C} \sum_{c=1}^{C} \mathbf{p}_c^2 - \frac{1}{C^2}. \tag{7}$$

We take the derivative of $v(\mathbf{p})$ w.r.t. τ, and obtain:

$$\begin{aligned} \frac{\partial v(\mathbf{p})}{\partial \tau} &= \frac{2}{C} \sum_{c=1}^{C} \mathbf{p}_c \frac{\partial \mathbf{p}_c}{\partial \tau} = \frac{2}{C\tau^2} \sum_{c=1}^{C} \mathbf{p}_c^2 \left(\sum_{j=1}^{C} \mathbf{p}_j \mathbf{f}_j - \mathbf{f}_c \right) \\ &= \frac{2}{C\tau^2} \left(\left(\sum_{c=1}^{C} \mathbf{p}_c^2 \right) \left(\sum_{j=1}^{C} \mathbf{p}_j \mathbf{f}_j \right) - \sum_{c=1}^{C} \mathbf{p}_c^2 \mathbf{f}_c \right). \end{aligned} \tag{8}$$

Substituting $\mathbf{f}_c = \tau \log \mathbf{p}_c + \tau \log Z$, we get:

$$\frac{\partial v(\mathbf{p})}{\partial \tau} = \frac{2}{C\tau} \left(\left(\sum_{c=1}^{C} \mathbf{p}_c^2 \right) \left(\sum_{j=1}^{C} \mathbf{p}_j \log \mathbf{p}_j \right) - \sum_{c=1}^{C} \mathbf{p}_c^2 \log \mathbf{p}_c \right). \tag{9}$$

Then, we define $\hat{\mathbf{p}}_c = \mathbf{p}_c^2 / \sum_{c=1}^{C} \mathbf{p}_c^2$, and we could derive the following equation:

$$\frac{\partial v(\mathbf{p})}{\partial \tau} = \frac{2 \left(\sum_{c=1}^{C} \mathbf{p}_c^2 \right)}{C\tau} \left(\sum_{c=1}^{C} \mathbf{p}_c \log \mathbf{p}_c - \sum_{c=1}^{C} \hat{\mathbf{p}}_c \log \mathbf{p}_c \right). \tag{10}$$

We then calculate the KL divergence of $\hat{\mathbf{p}}$ and $\mathbf{p}$:

$$KL(\hat{\mathbf{p}} || \mathbf{p}) = \sum_{c=1}^{C} \hat{\mathbf{p}}_c \log \frac{\hat{\mathbf{p}}_c}{\mathbf{p}_c} = \sum_{c=1}^{C} \hat{\mathbf{p}}_c \log \frac{\mathbf{p}_c}{\sum_j \mathbf{p}_j^2} = \sum_{c=1}^{C} \hat{\mathbf{p}}_c \log \mathbf{p}_c - \sum_{c=1}^{C} \hat{\mathbf{p}}_c \log \left(\sum_{j=1}^{C} \mathbf{p}_j^2 \right). \tag{11}$$

Due the non-negativity of KL divergence, we have:

$$\sum_{c=1}^{C} \hat{\mathbf{p}}_c \log \mathbf{p}_c \geq \sum_{c=1}^{C} \hat{\mathbf{p}}_c \log \left(\sum_{j=1}^{C} \mathbf{p}_j^2 \right) = \log \left(\sum_{c=1}^{C} \mathbf{p}_j^2 \right). \tag{12}$$

Hence, we have:

$$\sum_{c=1}^{C} \mathbf{p}_c \log \mathbf{p}_c - \sum_{c=1}^{C} \hat{\mathbf{p}}_c \log \mathbf{p}_c \leq \sum_{c=1}^{C} \mathbf{p}_c \log \mathbf{p}_c - \log \sum_{c=1}^{C} \mathbf{p}_c^2 \leq 0, \tag{13}$$

where the first inequality is according to Eq. 12 and the last inequality is according to Jensen's inequality, This proves that $\frac{\partial v(\mathbf{p})}{\partial \tau} \leq 0$. □

Proposition B.2 ((Proposition 4.3 in the Body))**.** *Under the Assumption 4.2,* $\mathbf{p}_y$ *monotonically decreases as* τ *increases, and* $e(\mathbf{q})$ *monotonically increases as* τ *increases. As* $\tau \to \infty$, $e(\mathbf{q}) \to 1/C$.

Proof. We take the derivative of $\mathbf{p}_y$ w.r.t. τ, and obtain:

$$\frac{\partial \mathbf{p}_y}{\partial \tau} = \frac{\mathbf{p}_y}{\tau^2} \left(\sum_{c=1}^{C} \mathbf{p}_c \mathbf{f}_c - \mathbf{f}_y \right) \leq 0, \tag{14}$$

which shows the monotonicity of $\mathbf{p}_y$ w.r.t. τ. Then the later is obvious because $e(\mathbf{q}) = \frac{1}{C-1} \sum_{c \neq y} \mathbf{p}_c = \frac{1}{C-1}(1 - \mathbf{p}_y)$. Finally, according to the properties of limitation, $\lim_{\tau \to \infty} \mathbf{p}_y = \lim_{\tau \to \infty} \frac{\exp(\mathbf{f}_y/\tau)}{\sum_{j=1}^{C} \exp(\mathbf{f}_j/\tau)} = 1/C$. □

Proposition B.3 (Derived Variance vs. Inherent Variance (Proposition 4.4 in the Body))**.** *The derived variance is determined by the square of derived average and the inherent variance via the following equation:*

$$\underbrace{v(\mathbf{q})}_{DV} = (C-1)^2 \underbrace{e^2(\mathbf{q})}_{DA^2} \underbrace{v(\tilde{\mathbf{q}})}_{IV}. \tag{15}$$

Proof. With the property in Eq. 3, we have: $\tilde{\mathbf{q}}_{c'} = \frac{\exp(\mathbf{g}_{c'}/\tau)}{\sum_j \exp(\mathbf{g}_j/\tau)} = \frac{\mathbf{p}_c}{\sum_{j \neq y} \mathbf{p}_j} = \frac{\mathbf{p}_c}{1 - \mathbf{p}_y}$. c' is the corresponding index of $\mathbf{p}_c$ in $\mathbf{q}$ after removing the correct class probability $\mathbf{p}_y$, i.e., $c' = c$ when $c < y$, and $c' = c - 1$ when $c > y$.

$$\begin{aligned} v(\mathbf{q}) &= \frac{1}{C-1} \sum_{c \neq y} \mathbf{p}_c^2 - e^2(\mathbf{q}) = \frac{1}{C-1} \sum_{c \neq y} \mathbf{p}_c^2 - \frac{(1-\mathbf{p}_y)^2}{(C-1)^2} \\ &= (1-\mathbf{p}_y)^2 \left(\frac{1}{C-1} \sum_{c \neq y} \left(\frac{\mathbf{p}_c}{1 - \mathbf{p_y}} \right)^2 - \frac{1}{(C-1)^2} \right) \\ &= (1-\mathbf{p}_y)^2 \left(\frac{1}{C-1} \sum_{c'} \tilde{\mathbf{q}}_{c'}^2 - \frac{1}{(C-1)^2} \right) \\ &= (1-\mathbf{p}_y)^2 v(\tilde{\mathbf{q}}) = (C-1)^2 e^2(\mathbf{q}) v(\tilde{\mathbf{q}}). \end{aligned} \tag{16}$$

□

C Details of Datasets, Networks and Training

C.1 Dataset Details

The datasets used in our experiments are CIFAR-10/CIFAR-100 (C10/C100) [21], TinyImageNet (TIN) [43], CUB [47], Stanford Dogs (Dogs) [19], and Google Speech Commands (GSC) [48].

CIFAR-10/CIFAR-100 are image classification datasets, and each contains 50K training and 10K test samples of size 32 × 32. TinyImageNet contains 200 classes, 100K training samples, and 10K

Table 4: Statistics of datasets. **Task**: Image Classification (IC), Fine-Grained Recognition (FGR), Keyword Spotting (KWS). **Size**: the input size of a single sample, we omit the dimension of channel. **Networks**: ResNet (RES), WideResNet (WRN), ResNeXt (RNX), ShuffleNetV1/2 (SFV1/2), MobileNetV2 (MV2), AlexNet (ANet).

Dataset	Task	Size	C	Num.Tr	Num.Te	Teacher Networks	Student Networks
C10	IC	(32, 32)	10	50K	10K	RES, WRN, RNX	VGG8, SFV1, MV2
C100	IC	(32, 32)	100	50K	10K	RES, WRN, RNX	VGG8, SFV1, MV2
TIN	IC	(64, 64)	200	100K	10K	WRN50-2	ANet, SFV2, MV2
CUB	FGR	(224, 224)	200	6K	5.8K	RNX101-32-8d	ANet, SFV2, MV2
Dogs	FGR	(224, 224)	120	12K	8.6K	RNX101-32-8d	ANet, SFV2, MV2
GSC	KWS	(101, 40)	35	106K	11K	RES	DSCNN

Table 5: Details of networks and their performances on training and test set (Accs). "RNX" is abbreviated as "R" to save space. Student networks are shaded while teachers are not.

Network	N.P	Accs	Network	N.P	Accs	Network	N.P	Accs
CIFAR-10 Teachers								
ResNet			WideResNet			ResNeXt		
RES14	0.18M	98.4, 91.5	WRN28-1	0.37M	99.7, 92.8	R29-4-4d	1.2M	100, 93.9
RES20	0.27M	99.5, 92.3	WRN28-2	1.5M	100, 94.9	R29-8-4d	1.7M	100, 94.7
RES32	0.47M	99.9, 93.5	WRN28-3	3.3M	100, 95.3	R29-16-4d	2.7M	100, 95.2
RES44	0.66M	99.9, 93.8	WRN28-4	5.8M	100, 95.6	R29-24-4d	3.8M	100, 95.3
RES56	0.86M	100, 93.9	WRN28-6	13M	100, 95.9	R29-32-4d	4.8M	100, 95.6
RES110	1.7M	100, 94.3	WRN28-8	23M	100, 96.1	R29-64-4d	8.9M	100, 95.8
CIFAR-10 Students								
VGG8	3.9M	99.9, 91.7	SFV1	0.86M	99.9, 92.1	MV2	0.7M	92.6, 88.4
CIFAR-100 Teachers								
ResNet			WideResNet			ResNeXt		
RES14	0.18M	81.1, 67.0	WRN28-1	0.38M	91.8, 70.0	R29-4-4d	1.3M	99.8, 75.1
RES20	0.28M	88.1, 69.1	WRN28-2	1.5M	99.8, 74.2	R29-8-4d	1.8M	99.9, 76.0
RES32	0.47M	94.3, 71.1	WRN28-3	3.3M	100, 76.7	R29-16-4d	2.8M	100, 77.9
RES44	0.67M	97.3, 72.2	WRN28-4	5.9M	100, 77.9	R29-24-4d	3.8M	100, 79.1
RES56	0.86M	98.7, 72.9	WRN28-6	13M	100, 79.1	R29-32-4d	4.9M	100, 79.3
RES110	1.7M	99.8, 74.1	WRN28-8	23M	100, 79.7	R29-64-4d	8.9M	100, 79.9
CIFAR-100 Students								
VGG8	4.0M	99.8, 69.9	SFV1	0.95M	99.9, 70.0	MV2	0.81M	83.7, 64.8
TinyImageNet Teachers								
-	-	-	WRN50-2	67M	100, 66.3	-	-	-
TinyImageNet Students								
ANet	2.7M	98.5, 34.6	SFV2	1.5M	94.3, 45.8	MV2	2.5M	85.9, 52.0
CUB Teachers								
-	-	-	-	-	-	R101-32-8d	87M	98.1, 79.5
CUB Students								
ANet	2.7M	92.8, 55.7	SFV2	1.5M	93.7, 71.2	MV2	2.5M	95.8, 74.5
Stanford Dogs Teachers								
-	-	-	-	-	-	R101-32-8d	87M	97.8, 74.0
Stanford Dogs Students								
ANet	2.6M	88.0, 50.2	SFV2	1.4M	92.6, 68.7	MV2	2.4M	94.5, 68.7
Google Speech Commands Teachers								
RES	1.9M	98.8, 99.2	-	-	-	-	-	-
Google Speech Commands Students								
DSCNN	59.8K	93.2, 94.5	-	-	-	-	-	-

test samples of size 64×64. CUB and Stanford Dogs are fine-grained recognition datasets, which correspondingly contain 200 and 120 classes. Google Speech Commands is a benchmark for keyword spotting [60, 41], which usually needs efficient model deployment. It aims to identify whether a 1s-long speech recording is a word, silence, or unknown. There are total 35 classes (35 words) in this task. We extract 40 MFCC features for each 30ms window frame with a stride of 10ms. We also follow the settings in Google Speech Commands (GSC) [48]: performing random time-shift of $Y \in [-100, 100]$ milliseconds and adding 0.1 volume background noise with a probability of 0.8. The details are listed in Tab. 4. C denotes the number of classes. "Num.Tr" and "Num.Te" denote the number of training and test samples.

C.2 Network Details

For teacher networks, we use different versions of ResNet [11], WideResNet [56], ResNeXt [49]. For student networks, we use VGG [39], ShuffleNetV1/V2 [59, 29], AlexNet [22], MobileNetV2 [37], and DSCNN [60].

- **ResNet (RES)**: We utilize ResNet for CIFAR-10/100 and Google Speech Commands as teachers. For CIFAR-10/100, we use the original ResNet versions with "6n+2" layers in [11]. These ResNets only use the "basic block" for CIFAR data. Specifically, we vary the number of layers in $\{14, 20, 32, 44, 56, 110\}$. For Google Speech Commands, we utilize the ResNet version proposed in [41]. We modify the number of layer as 15 and the channel as 128.
- **WideResNet (WRN)**: We utilize WideResNet for CIFAR-10/100 and TinyImageNet as teachers. For CIFAR-10/100, we use the WideResNet proposed in [56]. We keep the depth as 28 and vary the widen factors in $\{1, 2, 3, 4, 6, 8\}$. For TinyImageNet, we use the WideResNet50-2 provided in PyTorch [2].
- **ResNeXt (RNX)**: We utilize ResNeXt for CIFAR-10/100, CUB and Stanford Dogs as teachers. For CIFAR10-/100, we utilize the original ResNeXt versions for CIFAR data as proposed in [49]. We take 29 layers and set the base width as 4, varying the number of groups in $\{4, 8, 16, 24, 32, 64\}$. For CUB and Stanford Dogs, we use the ResNeXt101-32-8d provided in PyTorch.
- **VGG8**: We utilize VGG8 [39] for CIFAR-10/100 as the student. Although it contains more parameters, the plain architecture makes it weaker.
- **ShuffleNetV1/2 (SFV1/2)**: We utilize ShuffleNet for CIFAR-10/100, TinyImageNet, CUB, and Dogs as students. For CIFAR-10/100, we use ShuffleNetV1 [59] provided in RepDistiller repository [3]. For other datasets, we use ShuffleNetV2 [29] provided in PyTorch.
- **MobileNetV2 (MV2)**: We utilize MobileNet for CIFAR-10/100, TinyImageNet, CUB, and Dogs as students. For CIFAR-10/100, we use MobileNetV2 [37] in RepDistiller repository. For other datasets, we use ShuffleNetV2 [37] provided in PyTorch.
- **AlexNet (ANet)**: We utilize AlexNet provided in PyTorch for TinyImageNet, CUB, and Dogs as students.
- **DSCNN**: We use DSCNN [60] for Google Speech Commands as the student. It utilizes depth-separable convolution to accelerate training and inference. We set the basic number of channels as 96.

Additionally, for CUB and Stanford Dogs, we use the corresponding pre-trained models provided in PyTorch as initialization. The number of parameters ("N.P") and the training and test accuracies are in Tab. 5.

C.3 Training Details

Train Teachers We first train teacher networks on corresponding datasets and then save the checkpoints of them.

Train Students We train students via the loss function in Eq. 1. We set $\lambda = 0.5$ by default and tune $\tau \in \{1.0, 2.0, 4.0, 8.0, 12.0, 16.0\}$. Additionally, we vary the student's τ in two settings: (1) the same

[2] `https://pytorch.org/vision/stable/models.html`

[3] `https://github.com/HobbitLong/RepDistiller/tree/master/models`

Table 6: Comparisons with SOTA methods on CIFAR-10. ResNet110, WRN28-8, and RNX29-64-4d are used as teachers. VGG8, SFV1, and MV2 are students.

Teacher	ResNet110 (94.3)			WRN28-8 (96.1)			RNX29-64-4d (95.8)			Avg
Student	VGG8	SFV1	MV2	VGG8	SFV1	MV2	VGG8	SFV1	MV2	
NoKD	91.68	92.11	88.36	91.68	92.11	88.36	91.68	92.11	88.36	90.72
ST-KD	92.47	93.20	**88.83**	92.48	93.19	**88.72**	92.56	93.20	88.70	91.48
KD	92.29	92.80	88.60	92.35	93.07	88.54	92.44	93.18	88.37	91.29
ESKD	92.07	92.84	88.18	92.06	92.84	88.33	92.32	92.90	88.21	91.08
TAKD	92.10	93.13	88.36	92.33	92.59	88.15	92.31	93.03	88.50	91.17
SCKD	91.75	92.79	88.70	91.66	92.57	88.50	91.91	92.59	88.30	90.97
KD+ATS	**92.88**	**92.99**	88.63	**92.84**	**93.20**	88.43	**92.73**	**93.27**	**88.96**	**91.55**

as the teacher's; (2) 1.0. For our proposed KD method, the pair of (τ_1, τ_2) in Eq. 5 is searched in $\{(2.0, 1.0), (3.0, 1.0), (3.0, 2.0), (4.0, 2.0), (4.0, 3.0), (5.0, 2.0)\}$. For our method, we keep the student's temperature as 1.0 by default.

Compared Methods We then explain the compared methods in Tab. 2 and Tab. 3. For CIFAR-100/CIFAR-10, we use ResNet110, WRN28-8, and RNX29-64-4d as teachers. For TinyImageNet, CUB, and Stanford Dogs, we use WRN50-2 and ResNeXt101-32-8d as teachers. These models are complex networks with a larger depth, or a larger widen factor, or more groups. We have also explored the standard deviation of these KD learning methods. KD performances are relatively stable among several training random seeds with the same hyper-parameter settings. The standard deviations are also nearly the same across different KD methods, which are about 0.1%. Hence, we do not report the standard deviations due to space limitation.

- **NoKD**: training the student without distillation.
- **ST-KD**: training the student under the guidance of a smaller teacher. For CIFAR-100, we select the best performance from all smaller networks as shown in Fig. 9. For TinyImageNet, CUB, and Stanford Dogs, we correspondingly use WRN26-2, RNX50-32-4d, RNX50-32-4d as smaller teachers.
- **KD**: training the student under the guidance of the larger teacher.
- **ESKD** [6]: training the student under the guidance of the *early-stopped teacher*. Specifically, we train the larger teachers only for $\{60, 120, 150\}$ epochs using the cosine annealing learning rate and report the best accuracy of the student.
- **TAKD** [31]: training the student under the guidance of the *teacher assistant*. Specifically, we vary the TA in {ResNet20, ResNet44} for ResNet110, {WRN28-2, WRN28-4} for WRN28-8, {RNX29-8-4d, RNX29-8-24d} for RNX29-64-4d, {WRN26-2} for WRN50-2, and {RNX50-32-4d} for RNX101-32-8d.
- **SCKD** [51]: training the student under the *student customized teacher*. We use two KD losses, including the KL divergence of probabilities (Eq. 1) and the L2 feature distance loss as done in [36].
- **Ens**: training two students with different initializations two times and taking their ensemble.
- **ResKD**: training the student first and then training the *residual student* to fit the residual. The ensemble of these two students is used for inference.
- **KD+ATS**: training the student with the larger teacher's guidance via the proposed ATS.
- **KD+ATS+Ens**: repeating the above two times and taking the ensemble results. When repeating "KD+ATS" two times, we use the same hyperparameters and keep the fairness of comparisons.

C.4 Figure Details

We explain some figures in detail.

Table 7: Comparisons with other KD methods on CIFAR-100. The results of first five columns are cited from [44]. The gray area shows the results of scenes with smaller "capacity gap". ReKD denotes ReviewKD [4].

Scene	Teacher	Student	KD	CC	NST	CRD	IE-AT	ReKD	Ours
ResNet110 → ResNet20	74.31	69.06	70.67	69.48	69.53	71.14	71.34	71.37	**71.57**
VGG13 → VGG8	74.64	70.36	72.98	70.71	71.53	73.94	**74.05**	73.60	73.65
ResNet56 → ResNet20	72.34	69.06	70.66	69.63	69.60	**71.16**	70.87	70.95	70.99
WRN40-2 → WRN16-2	75.61	73.26	74.92	73.56	73.68	75.48	74.80	**75.67**	75.03

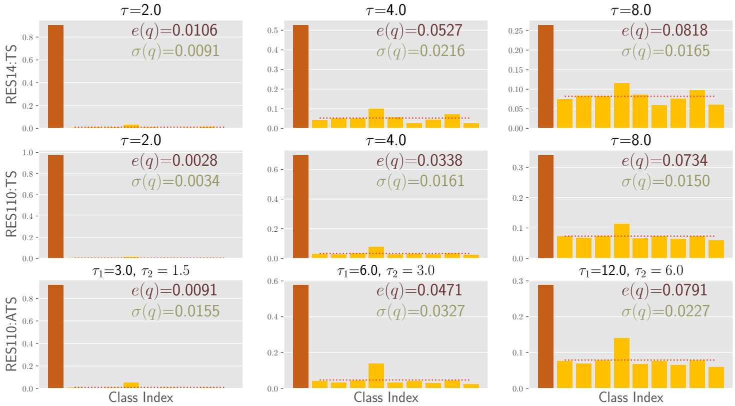

Figure 12: Probability vector visualization of a randomly selected training sample in CIFAR-10. The networks are various ResNet. The target class is $y = 1$. The bottom row shows the efficacy of ATS for a larger teacher.

- Fig. 2: for "ResNet→SFV1", we use ResNet14 as the small teacher ("ST") and ResNet110 as the large teacher ("LT"); for "WRN→MV2", we use WRN28-1 as the small teacher ("ST") and WRN28-8 as the large teacher ("LT").

- Fig. 3: on CIFAR-100, we vary 18 teachers and 3 students as listed in Tab. 5; we also vary temperatures in $\{1.0, 2.0, 4.0, 8.0, 16.0\}$; the total pairs of distillation combinations are $18 \times 3 \times 5 = 270$; for each pair, we calculate the teacher's *derived average* and *derived variance* averaged across 50K training samples; the KD improvement ratio is calculated by $(\text{Acc}_{\text{KD}} - \text{Acc}_{\text{NoKD}})/\text{Acc}_{\text{NoKD}}$.

- Fig. 6 and Fig. 5: the former calculates $\mathbf{f}_y$ (i.e., target logit) and $\sigma(\mathbf{g})$ (i.e., standard deviation of wrong logits) across training samples and plot the bins; the latter calculates $\sigma(\mathbf{q})$ (i.e., derived variance) and $\sigma(\tilde{\mathbf{q}})$ (i.e., inherent variance) and only report their mean and standard deviation across training samples.

- Fig. 4: for each pair of two teachers T_1 and T_2, we obtain their softened labels on all training samples; for each sample, we calculate the four metrics and report the average across training samples.

- Fig. 7: given a temperature τ, we could obtain softened probabilities of all training samples; then we calculate $\sigma(\mathbf{q})$ (i.e., derived variance) and $\sigma(\tilde{\mathbf{q}})$ (i.e., inherent variance) across all training samples and only report the average results.

- Fig. 9: For TS, we tune $\tau \in \{1.0, 2.0, 4.0, 8.0, 12.0, 16.0\}$ and vary the student's τ in two settings: (1) the same as the teacher's, (2) 1.0; For ATS, the pair of (τ_1, τ_2) in Eq. 5 is searched in $\{(2.0, 1.0), (3.0, 1.0), (3.0, 2.0), (4.0, 2.0), (4.0, 3.0), (5.0, 2.0)\}$; through adjusting the hyper-parameters and selecting the best results, we plot the performance curves.

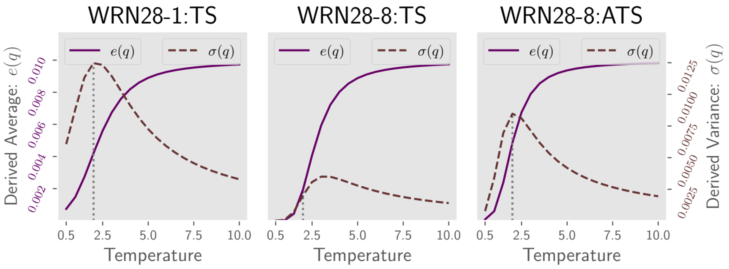

Figure 13: The change of *derived average* ($e(\mathbf{q})$) and *derived variance* ($v(\mathbf{q})$) as τ increases from 0.1 to 10.0 on CIFAR-100 using various WRN.

Table 8: Comparisons on Google Speech Commands under various types of KD. +/− denotes multiplying τ^2 or not in Eq. 1.

(λ, +/−)	(0.5, +)	(0.9, +)	(0.5, −)	(0.9, −)
KD+TS	95.52	95.37	94.55	94.37
KD+ATS	96.79	97.21	96.23	95.84

D More Experimental Studies

D.1 CIFAR-10 Results

We also compare our proposed methods with SOTA on CIFAR-10, and the results are listed in Tab. 6. Because CIFAR-10 only contains 10 classes and is an slightly simpler benchmark, the performance improvement is not so obvious as other datasets.

D.2 Speech Data Results

ATS could also perform well on speech data, i.e., the Google Speech Commands benchmark as shown in Tab. 8, where we vary several types of KD loss. Our proposed ATS does not depend on multiplying τ^2 or not in Eq. 1.

D.3 Comparisons With Other KD Methods on CIFAR-100

We also compare our proposed method with other KD methods, such as [44, 17, 34, 4, 16]. The previous work [44] provides a comprehensive experimental study on many KD methods. For fair comparison, we directly cite their results and only report several scenes of "Larger Teacher → Smaller Student", i.e., "ResNet110 (74.31%) → ResNet20 (69.06%)" and "VGG13 (74.64%) → VGG8 (70.36%)". For IE-KD [16], we use the proposed variant of IE-AT. For IE-KD [16] and ReviewKD [4], we utilize the public code that the authors provide. We also compare with these two methods because they are the recently proposed SOTA KD methods. The results are listed in Tab. 7. Consistently, our method could improve the KD up to 1% and the results are comparable with CRD that utilizes more advanced techniques. We also apply ATS to the cases that students have almost the same capacity as teachers, e.g., "WRN40-2 (75.61%) → WRN16-2 (73.26%)" and "ResNet56 (72.34%) → ResNet20 (69.06%)". Under these scenes, our method becomes not so effective and the performance improvement over KD seems more like the benefits of hyper-parameter tuning. Hence, our proposed ATS may be more effective when applying a more complex teacher to guide the learning process of a smaller student.

D.4 More Results for the Observations

We also present additional results for the observations in Fig. 8 and Fig. 7 to verify that the observations and conclusions are not accidental. Similar results are shown in Fig. 12 and Fig. 13.