UNDERSTANDING KNOWLEDGE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge distillation (KD), transferring knowledge from a cumbersome teacher model to a lightweight student model, has been investigated to design efficient neural architectures with high accuracy with a few parameters. However, there is a very limited understanding of why and when KD works well. This paper reveals KD's intriguing behaviors, which we believe useful in a better understanding of KD. We first investigate the role of the temperature scaling hyperparameter in KD. It is theoretically shown that the KD loss focuses on the logit vector matching rather than the label matching between the teacher and the student as the temperature grows up. We also find that KD with a sufficiently large temperature outperforms any other recently modified KD methods from extensive experiments. Based on this observation, we conjecture that the logit vector matching is more important than the label matching. To verify this conjecture, we test an extreme logit learning model, where the KD is implemented with Mean Squared Error (MSE) between the student's logit and the teacher's logit. The KD with MSE consistently shows the best accuracy for various environments. We analyze the different learning behavior of KD with respect to the temperature using a new data uncertainty estimator, coined as Top Logit Difference (TLD). We then study the KD performances for various data sizes. When there are a few data or a few labels, very interestingly, the incapacious teacher with a shallow depth structure facilitates better generalization than teachers having wider and deeper structures.

1 INTRODUCTION

Despite the considerable success of the deep neural networks in various tasks such as image classification and natural language processing, there have been increasing demands of building resource-efficient deep neural networks (e.g., fewer parameters) without sacrificing accuracy. Recent progress in this direction has involved in designing efficient neural architecture families (Howard et al., 2017; Tan & Le, 2019; Howard et al., 2019), sparsely training a network (Frankle & Carbin, 2018; Mostafa & Wang, 2019), quantizing the weight parameters (Banner et al., 2018; Zhao et al., 2019), and distilling knowledge from a well-learned network into another network (Hinton et al., 2015; Zhou et al., 2019).

The last of these, knowledge distillation (KD), is one of the most potent model compression techniques by transferring knowledge from a cumbersome model to a single small model (Hinton et al., 2015). KD utilizes the "soft" probabilities of a large "teacher" network instead of the "hard" targets (i.e., one-hot vectors) to train a smaller "student" network. Some studies have attempted to distill the hidden feature vector of the teacher network in addition to the soft probabilities so that the teacher can transfer rich information (Romero et al., 2014; Zagoruyko & Komodakis, 2016a; Srinivas & Fleuret, 2018; Kim et al., 2018; Heo et al., 2019b;a). KD method can be leveraged to reduce the generalization errors in teacher models (i.e., self-distillation; SD) (Zhang et al., 2019; Park et al., 2019) as well as model compression. In the generative models, a generator can be compressed by distilling the latent feature from a cumbersome generator (Aguinaldo et al., 2019).

Despite the increasing demands of KD, much is still a lack of understanding about why and when KD should work. In particular, Tian et al. (2019) argue that even original KD (Hinton et al., 2015) can outperform various other KD methods that distill the hidden feature vector (Table 1). To uncover several mysteries of KD, this paper attempts to shed light upon the behavior of neural networks trained with KD while the amount of distilled knowledge changes. Our contributions are summarized as follows:

Table 1: **Test accuracy of various KD methods** on CIFAR-100. 'WRN' indicates a family of Wide-ResNet. All student models share the same teacher model as WRN-28-4. SKD (Standard KD) and FKD (Full KD) represent the KD method (Hinton et al., 2015) with different hyperparameter values (α, τ) used in Eq. (1) - (0.1, 5) and (1.0, 20), respectively. MSE represents the KD with L2 regression loss between logits; see Appendix A for citations of other methods. Others are results reported in Heo et al. (2019a). Baseline indicates the model trained without teacher model.

Student	Baseline	SKD	FitNets	AT	Jacobian	FT	AB	Overhaul	FKD	MSE
WRN-16-2	72.68	73.53	73.70	73.44	73.29	74.09	73.98	75.59	**75.76** (↑)	75.54
WRN-16-4	77.28	78.31	78.15	77.93	77.82	78.28	78.64	78.20	78.84	**79.03** (↑)
WRN-28-2	75.12	76.57	76.06	76.20	76.30	76.59	76.81	76.71	**77.28** (↑)	**77.28** (↑)

- We conduct vast experiments considering the combination of teacher and student and two hyperparameters in the KD loss referring to Koratana et al. (2019). We observe that KD loss utilizing only the teacher's calibrated softmax output (Guo et al., 2017a) brings better performance than other KD settings (Table 1).
- We demonstrate that the original KD with high-temperature (i.e., a hyperparameter for calibration of neural networks) without ground-truth labels acts as the L2 regression of the logit (i.e., the input of softmax function) and shows almost the best results for many cases.
- Based on the second contribution, we test the L2 regression loss (i.e., Mean Squared Error; MSE) between the student's logit and the teacher's logit. Perhaps surprisingly, KD with the MSE loss outperforms other KD algorithms (Table 1).
- We propose a novel estimator of data uncertainty, referred to as **Top Logit Difference (TLD)**, based on the difference between the largest and the ground truth element in logit. TLD provides intuition regarding how knowledge transfer differs between students distilled from the softmax output with and without calibration (Guo et al., 2017a).
- We show that the benefit of KD depends on the data size. When a model is trained on a small portion of the full training data set, KD surpasses the vanilla SGD training. In terms of the teacher size, simple teachers, having fewer parameters than the student (i.e., knowledge expansion; KE (Xie et al., 2020)), show better accuracies than more massive teachers. These findings are the same in OOD prediction tasks.

1.1 Preliminary: Knowledge Distillation

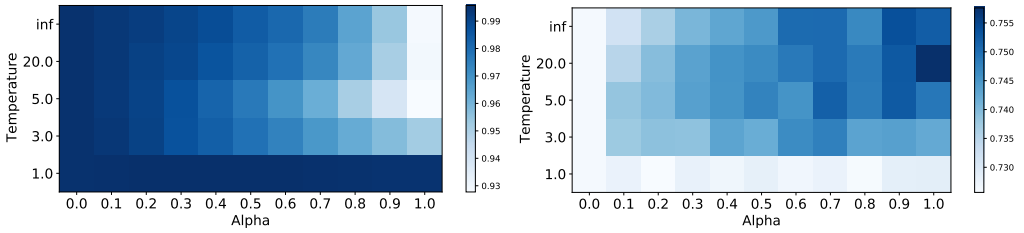
We provide a mathematical description of KD before introducing our study. Let us denote the softened probability vector in a network f as $\boldsymbol{p}^f(\boldsymbol{x};\tau) = \frac{e^{\boldsymbol{z}_k^f/\tau}}{\sum_j e^{\boldsymbol{z}_j^f/\tau}}$ where $\boldsymbol{x}$ is an input, τ is a temperature scaling hyperparameter (Guo et al., 2017a), and $\boldsymbol{z}_k^f$ is the value of a logit vector at index k. Then, the typical loss $\mathcal{L}$ for the student network is a linear combination of the cross entropy (CE) loss $\mathcal{L}_{CE}$ and the KD loss $\mathcal{L}_{KD}$:

$$\begin{aligned} &\mathcal{L} = (1-\alpha)\mathcal{L}_{CE}(p^s(\boldsymbol{x};1), q(\boldsymbol{x})) + \alpha\mathcal{L}_{KD}(p^s(\boldsymbol{x};\tau), p^t(\boldsymbol{x};\tau)), \\ &\text{where} \quad \mathcal{L}_{KD}(p^s(\boldsymbol{x};\tau), p^t(\boldsymbol{x};\tau)) = \tau^2 \sum_j p_j^t(\boldsymbol{x};\tau) \log \frac{p_j^t(\boldsymbol{x};\tau)}{p_j^s(\boldsymbol{x};\tau)} \end{aligned} \tag{1}$$

where s indicates the student network, t indicates the teacher network, $q(\boldsymbol{x})$ is a one-hot label vector of sample $\boldsymbol{x}$, and α is a hyperparameter of the linear combination. Standard choices are $\alpha = 0.1$ and $\tau \in \{3, 4, 5\}$.

1.2 Experimental Setup

In this paper, we use an experimental setup similar to the Heo et al. (2019a), Cho & Hariharan (2019), and Zhou et al. (2019): image classification on CIFAR-10, CIFAR-100, and ImageNet with a family of ResNet (RN) (He et al., 2016a) as well as that of Wide-ResNet (WRN) (Zagoruyko & Komodakis, 2016b) and machine translation on WMT14 En-De with the autoregressive transformer



(a) Training accuracy.

(b) Test accuracy.

Figure 1: **Grip maps of accuracies according to the change of α and τ** on CIFAR-100 when (teacher, student) = (WRN-28-4, WRN-16-2). It presents the grid maps of (a) **training top1 accuracies** and (b) **test top1 accuracies**. $\mathcal{L}_{KD}$ with $\tau = \infty$ is implemented with the hand-crafted gradient (Eq. (3)). Detailed values are in subsection B.3.

(AT) and the non-auto regressive transformer (NAT). We use a standard PyTorch SGD optimizer with momentum 0.9 and weight decay and apply standard data augmentation. Other than those mentioned, the training settings covered in the original papers (Heo et al., 2019a; Cho & Hariharan, 2019; Zhou et al., 2019) are used.

2 Temperature Scaling Hyperparameter τ of $\mathcal{L}_{kD}$

In this section, we conduct vast experiments and systematically break down the effects of temperature scaling hyperparameter τ in $\mathcal{L}_{KD}$ based on theoretical and empirical results. We further demonstrate that $\mathcal{L}_{KD}$ with infinite τ can be understood as the biased L2 regression.

We empirically observe that a generalization error of a student model becomes as less as that of the teacher when α for $\mathcal{L}_{KD}$ and τ in $\mathcal{L}_{KD}$ increase. As Figure 1 depicts, there are consistent tendencies that the higher the α and τ, the less over-fitting problem (i.e., less difference between training accuracy and test accuracy) (Figure 1) under the condition that τ is greater than 1. We find this consistency in various pairs of teachers and students (refer to the Appendix).

An intriguing effect of the hyperparameter τ in $\mathcal{L}_{KD}$ is that the student model attempts to imitate the logit distribution of the teacher model as τ goes to ∞, while the learning depends more on the classification outcomes of the teacher and the student as τ goes to 0. Here, we extend the gradient analysis of logit in Hinton et al. (2015) a little further. Consider the gradient of $\mathcal{L}$ in Eq. (1) *w.r.t.* logit z_k^s on each training instance:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{z}_k^s} = (1-\alpha)\frac{\partial \mathcal{L}_{CE}}{\partial \boldsymbol{z}_k^s} + \alpha \frac{\partial \mathcal{L}_{KD}}{\partial \boldsymbol{z}_k^s} = (1-\alpha)(p_k^s(\boldsymbol{x};1) - q_k(\boldsymbol{x})) + \alpha\tau(p_k^s(\boldsymbol{x};\tau) - p_k^t(\boldsymbol{x};\tau)) \quad (2)$$

where $q_k(\boldsymbol{x})$ is 1 if k is ground-truth class of sample $\boldsymbol{x}$, otherwise 0. The following theorem characterizes the tendency of the student models as τ changes.

Theorem 1 *Let K be the number of classes in the dataset, and $\mathbf{1}[\cdot]$ be the indicator function, which is 1 when the statement inside the bracket is true and 0 otherwise. Then,*

$$\lim_{\tau\to\infty} \frac{\partial \mathcal{L}_{KD}}{\partial \boldsymbol{z}_k^s} = \frac{1}{K^2}\sum_{j=1}^{K}\left((\boldsymbol{z}_k^s - \boldsymbol{z}_j^s) - (\boldsymbol{z}_k^t - \boldsymbol{z}_j^t)\right) \quad (3)$$

$$\lim_{\tau\to 0} \frac{1}{\tau}\frac{\partial \mathcal{L}_{KD}}{\partial \boldsymbol{z}_k^s} = \mathbf{1}_{[\arg\max_j \boldsymbol{z}_j^s = k]} - \mathbf{1}_{[\arg\max_j \boldsymbol{z}_j^t = k]} \quad (4)$$

Theorem 1 can explain the consistent tendency as follows: in the course of regularizing the $\mathcal{L}_{KD}$ with sufficiently large τ, the student model attempts to imitate the logit distribution of a teacher model. Specifically, the larger the τ, the more the $\mathcal{L}_{KD}$ makes the element-wise difference of the student's logit vector similar to that of the teacher (i.e., *logit vector matching*). On the other hand, when τ is close to 0, the gradient of $\mathcal{L}_{KD}$ does not consider the logit distributions but just identify

if the student and the teacher share the same output (i.e., *label matching*), which transfers limited information. Besides, there is a scaling issue when τ goes to 0. As decreasing τ, $\mathcal{L}_{KD}$ increasingly loses its qualities and eventually becomes less involved in the learning. One can easily fix the scaling issue by multiplying $1/\tau$ to $\mathcal{L}_{KD}$ for $\tau \leq 1$. The details are in Appendix B.

$\mathcal{L}_{KD}$ with infinite τ can be understood as the biased L2 regression as follows:

$$\lim_{\tau \to \infty} \nabla_{\boldsymbol{z}^s} \mathcal{L}_{KD} = \frac{1}{K}\left(\boldsymbol{z}^s - \boldsymbol{z}^t\right) + b \cdot \mathbb{1} \tag{5}$$

where b is $-\frac{1}{K^2}\sum_{j=1}^{K}\left(\boldsymbol{z}_j^s - \boldsymbol{z}_j^t\right)$ and $\mathbb{1}$ is a vector whose elements are equal to one.

As derived in Eq. (5), the gradient *w.r.t.* logit $\boldsymbol{z}^s$ can be seen as the biased value of $(\boldsymbol{z}^s - \boldsymbol{z}^t)$. Because the softmax output is the same even if all elements of logit increase equally, $\mathcal{L}_{KD}$ with $\tau = \infty$ may be replaced with L2 regression (MSE). On the other side, L2 regression enables the student to learn the teacher's energy function as well without loss of logit summation (Grathwohl et al., 2019) while $\mathcal{L}_{KD}$ with $\tau = \infty$ does not. Grathwohl et al. (2019) showed that logit summation could be utilized as an energy function for training an energy-based model. From this perspective, MSE loss may transfer additional knowledge about the teacher's energy of each instance. Yet, in the experiment, no significant changes occur through this difference. As Table 2 shows, the larger τ, the higher test accuracy and MSE loss has similar performance with $\mathcal{L}_{KD}$ with $\tau = 20$ or ∞.

Table 2: Top1 test accuracies on CIFAR-100. WRN-28-4 is used as a teacher for $\mathcal{L}_{KD}$ and MSE when α=1.0 in $\mathcal{L}$.

Student	$\mathcal{L}_{CE}$	$\mathcal{L}_{KD}$					MSE
		τ=1	τ=3	τ=5	τ=20	τ=∞	
WRN-16-2	72.68	72.90	74.24	74.88	**75.76**	75.51	75.54
WRN-16-4	77.28	76.93	78.76	78.65	78.84	78.61	**79.03**
WRN-28-2	75.12	74.88	76.47	76.60	**77.28**	76.86	**77.28**
WRN-28-4	78.88	78.01	78.84	79.36	79.72	79.61	**79.79**
WRN-40-6	79.11	79.69	79.94	79.87	79.82	79.80	**80.25**

3 TLD AND TRAINING ACCURACY

$\mathcal{L}_{KD}$ with sufficiently large τ leads to better optima while training accuracy decreases (Figure 1). In this section, we seek to answer this question. To investigate this phenomenon, we propose a novel uncertainty estimator of each training instances as follows:

- **Top logit difference (TLD)**: It is defined as $\boldsymbol{z}^s_{k_*} - \boldsymbol{z}^s_{k_t}$, where k_* indicates the index of the true label and $k_t = \arg\max_{k:k\neq k_*} \boldsymbol{z}^s_k$ denotes the largest elements in logit z^s except k_*. The greater TLD, the more confidently and correctly predicted.

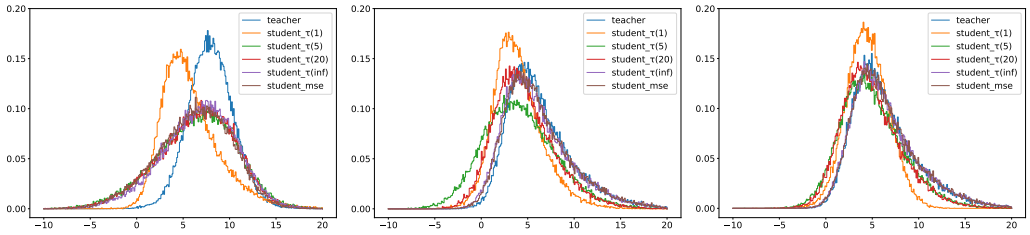
Recent studies (Tang et al., 2020; Yuan et al., 2020) demonstrated that the values except for a few high values in logit does not provide any information such as similarity information between categories in the course of distillation. In this respect, we consider the difference between the largest logit value and the ground-truth value as an estimator.

TLD contains not only the ground truth-related information but also the confidence of the teacher's prediction. Positive TLD implies the model predicts it correctly, while negative TLD is in the opposite. In addition, we empirically observe that the probability density function (pdf) of TLD seems to be bell-shaped (Figure 2), while the distribution of entropy on the model's softmax outputs, regardless of the calibration, is positively skewed towards 0. As Table 3 shows, though lower entropy generally means higher TLD, TLD and entropy are not perfectly aligned on the same side. In particular, the correlation gradually weakens as the model size (i.e., the number of parameters) increases.

Table 3: Pearson correlation coefficients (PCC) between entropy and TLD for each training instances. All models are trained with $\mathcal{L}_{CE}$ on CIFAR-100. The bold indicates p-value < 0.05.

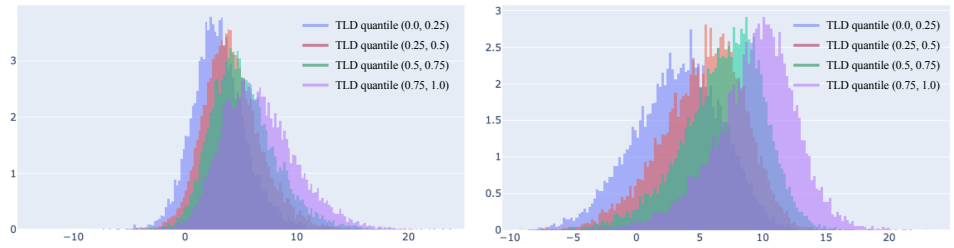
Model	WRN-28-4	WRN-16-4	WRN-28-2	WRN-16-2
PCC	$\mathbf{-0.3567}$	$\mathbf{-0.4590}$	$\mathbf{-0.4982}$	$\mathbf{-0.6950}$

To unravel the reason why training accuracy changes when $\mathcal{L}_{KD}$ with large τ applies, we visualize the pdf from the histogram of the TLD over the entire training data in various pairs of teacher and student (Figure 2): (1) the teacher has more parameters than the student (KD), (2) the teacher and the student share the same architecture (self-distillation; SD), and (3) the teacher has fewer parameters than the student (knowledge expansion; KE) (Xie et al., 2020). One striking difference among KD,



(a) KD: WRN-28-4 to WRN-16-2 (b) SD: WRN-16-2 to WRN-16-2 (c) KE: WRN-16-2 to WRN-28-4

Figure 2: Pdf of TLD. All students are trained with $\alpha = 1.0$ and all teachers do with $\mathcal{L}_{CE}$. WRN-16-2 student models share other training recipes such as learning rate, batch size, and weight decay.



(a) model trained with $\mathcal{L}_{CE}$ (b) model trained with MSE

Figure 3: Comparison of pdf of TLD between WRN-16-2 models trained with $\mathcal{L}_{CE}$ and with MSE (teacher: WRN-28-4). (a) and (b) has 4 different pdfs of TLD whose data belongs to different quantiles of teacher's pdf of TLD.

SD, and KE is that the TLD distribution of the student trained with SD or KE practically catches up that of the teacher (Figure 2b and 2c; $\tau = 20, \infty$ and MSE) while KD can not (Figure 2a). In theory, $\mathcal{L}_{KD}$ is minimized at the same solution regardless of the temperature, when the student has sufficient capacity to learn the exact logit of the teacher. Therefore, the TLD distributions are close to the teacher's TLD in SD and KE, where the student has a bigger structure than the teacher.

The student of KD, in contrast, does not have sufficient capacity to learn the teacher. Thus, the student's TLD pdfs seem quite variant from that of the teacher. Moreover, the TLD pdf of $\tau = 1$ is also very different from others. With $\tau = 1$, as derived in Eq. (4), the student seems to learn the teacher's predicted label more than the teacher's logit, whereas the student strives to match the teacher's logit distribution as τ gradually increases from Eq. (3).

To investigate the substantial differences at the example level, we further analyze the pdf of TLD of models trained with $\mathcal{L}_{CE}$ and with $\mathcal{L}_{KD}$ on four different bundles of data in consideration of the teacher's TLD values. For example, if the quantile is 0.1-0.8, then the training dataset is constructed with the data whose TLD values of the teacher model range from 10% and 80%. In Figure 3a and 3b, each color indicates different bundle of data whose data is in a particular quantile scope of teacher's TLD values. As Figure 3a and 3b show, both the models trained with $\mathcal{L}_{CE}$ and with $\mathcal{L}_{KD}$ follow the same relative order with the teacher to capture the TLD distribution. However, the student trained with $\mathcal{L}_{KD}$ seems to consider the TLD order more than that of $\mathcal{L}_{CE}$ as each quantile is more clearly separated. This result confirms that $\mathcal{L}_{KD}$ forces a student to learn even the teacher's degree of data uncertainty at the instance level.

We also check the performance according to the different bundles of distilled data (Table 4). Since the neural networks have enough capacity to learn the training data, all the training accuracies are very high. Our results also show that the KD training accuracy is slightly lower than the corresponding CE result as KD tries to learn the logit as well. To understand data difficulty more clearly, we test the accuracy of the undistilled data that is not utilized for the training. Notably, both learning methods make more errors as the undistilled data set consists of smaller TLD values, verifying that TLD indicates difficulty. Perhaps interestingly, our results indicate that the test accuracy does not

Table 4: Top1 test accuracies of WRN-16-2 models trained with $\mathcal{L}_{CE}$ and with MSE (teacher: WRN-28-4) on CIFAR-100. We evaluate the accuracies of distilled training data, undistilled training data, and test data. Sampling is based on the quantile of pdf of TLD from WRN-28-4 trained with $\mathcal{L}_{CE}$.

type	Distilled training data				Undistilled training data				Test data			
quantile	0.0-0.7	0.1-0.8	0.2-0.9	0.3-1.0	0.0-0.7	0.1-0.8	0.2-0.9	0.3-1.0	0.0-0.7	0.1-0.8	0.2-0.9	0.3-1.0
CE	**99.72**	**99.84**	**99.93**	**99.88**	79.92	71.62	63.51	53.21	68.85	68.51	68.83	68.86
KD	92.11	94.10	95.70	96.54	**87.51**	**77.83**	**69.34**	**60.03**	**73.85**	**73.65**	**73.88**	**73.04**

Table 5: Comparison of top1 accuracies on various data sets with different pairs of teacher and student models. 'CE' indicates the model trained with $\mathcal{L}_{CE}$. All teacher models are trained with the dataset whose (δ, ζ) is equal to $(1.0, 1.0)$. We report the best result over 3 individual runs with different initializations.

data set	type	teacher	student	$\delta = 1.0, \zeta = 0.1$							$\delta = 0.1, \zeta = 1.0$						
				CE	τ=1.0	τ=3.0	τ=5.0	τ=20.0	$\tau = \infty$	MSE	CE	τ=1.0	τ=3.0	τ=5.0	τ=20.0	$\tau = \infty$	MSE
CIFAR-100	KD	WRN-28-4			44.62	53.70	56.77	60.15	60.21	**61.01**		8.96	9.19	9.42	10.96	**12.19**	11.53
	SD	WRN-16-4	WRN-16-4	43.29	44.01	54.44	59.27	64.72	64.72	**67.10**	9.05	9.17	9.38	10.04	15.30	24.71	**27.43**
	KE	WRN-16-2			51.63	61.91	64.12	**67.22**	65.41	67.03		10.38	18.53	26.84	43.74	42.49	**44.63**
	KD	WRN-28-4			42.28	48.25	51.60	54.93	56.08	**56.51**		9.02	9.21	9.15	9.69	**10.54**	9.93
	SD	WRN-28-2	WRN-28-2	40.58	42.47	52.44	53.45	59.69	59.86	**60.47**	8.92	9.09	9.39	9.50	14.37	**16.51**	16.43
	KE	WRN-16-2			44.69	57.62	59.15	63.38	63.45	**64.27**		9.18	11.05	14.72	**32.36**	31.39	31.38

data set	type	teacher	student	$\delta = 1.0, \zeta = 0.1$							$\delta = 0.2, \zeta = 1.0$						
				CE	τ=1.0	τ=3.0	τ=5.0	τ=20.0	$\tau = \infty$	MSE	CE	τ=1.0	τ=3.0	τ=5.0	τ=20.0	$\tau = \infty$	MSE
CIFAR-10	KD	WRN-28-4			81.26	84.11	84.24	**86.30**	85.99	86.24		19.84	19.97	21.01	24.72	**29.95**	29.23
	SD	WRN-16-4	WRN-16-4	80.41	80.94	84.63	85.30	88.15	88.15	**88.41**	19.88	19.87	21.50	29.09	37.25	50.91	**51.39**
	KE	WRN-16-2			80.60	85.35	86.71	88.46	89.03	**89.20**		19.88	27.65	38.08	57.65	**65.62**	65.28
	KD	WRN-28-4			79.65	80.22	81.88	**84.28**	80.72	83.63		19.80	19.82	19.93	20.40	**21.45**	20.37
	SD	WRN-28-2	WRN-28-2	79.86	80.23	81.14	83.46	84.65	84.67	**84.75**	19.89	19.72	19.80	21.49	25.29	33.03	**36.41**
	KE	WRN-16-2			80.79	84.05	85.13	87.10	**87.47**	87.01		19.87	22.83	30.55	45.46	**56.16**	54.03

depend on the different bundles, while KD is much better than CE. The correlation between the test accuracy and the training data difficulty is a promising topic for further research.

4 KD AND DATA SIZE

In this section, we study how the use of data size (i.e., the number of data) affects the generalization of a student model. Here, for data usage, the data is sampled in consideration of the balance between classes: (1) **class-balanced sampling:** the number of data for each class in the entire training is equally reduced to check if KD is robust to the amount of training data and (2) **class-imbalanced sampling:** some classes from the entire training are excluded to check if the model trained with KD can predict out-of-distribution (OOD) data correctly. Class-imbalanced sampling is similar to the experiment of Hinton et al. (2015). They addressed the student's ability to learn indirect class-information that exists in teacher learning, but do not exist in student learning using the MNIST dataset. They distilled 9 classes out of 10 classes, but it has not been dealt with the extremely class-imbalanced case. Here, we attempt to investigate the result of an extreme case.

To this aim, we handle the amount of training data with hyperparameters δ and ζ, where δ is the ratio of the number of classes sampled to the total number of classes and ζ is the ratio of the number of data sampled for actual training to the total number of the training dataset. For instance, when a δ is 0.1 on CIFAR-100, the training dataset is reconstructed to have only ten classes, and when a ζ is 0.1 on CIFAR-100, the training dataset is reconstructed to have only 50 samples for each class out of 500 samples. If both δ and ζ are equal to 1.0, the entire training dataset is used to train a model.

$\mathcal{L}_{KD}$ and MSE Table 5 shows the performance of student models according to the change of teacher model and $\mathcal{L}_{KD}$ on a few data or a few labels. Here, we observe the consistent tendency mentioned in section 2 that the larger τ, the higher the test accuracy, and MSE achieves similar performance with $\tau = 20$ or ∞. Based on this discovery, we set the hyperparameters α and τ to 1.0 and 20.0, respectively, in the following experiments.

Class-balanced sampling (δ=1.0 columns of Table 6) We observe that, when a few training data is applied ($\zeta \ll 1.0$ & δ=1.0), the model trained with $\mathcal{L}_{KD}$ performs significantly better than the

Table 6: Comparison of top1 accuracies on various data sets with different pairs of teacher and student models. 'CE' indicates the models trained with $\mathcal{L}_{CE}$. All teacher models are trained with the dataset whose (δ, ζ) is equal to $(1.0, 1.0)$. We report the best result over 3 individual runs with different initializations.

data set	type	teacher	student	$\zeta, \delta = 1.0$	$\delta = 1.0$					$\zeta = 1.0$				
					ζ=0.1	ζ=0.2	ζ=0.3	ζ=0.4	ζ=0.5	δ=0.1	δ=0.2	δ=0.3	δ=0.4	δ=0.5
CIFAR-100	CE	None	WRN-16-4	76.89	43.29	57.56	63.94	66.83	69.81	9.05	17.39	25.50	32.69	40.85
	KD	WRN-28-4		**78.84**	60.15	68.52	71.68	73.63	75.54	10.96	24.02	32.17	40.76	49.02
	SD	WRN-16-4		77.15	64.72	**72.81**	**74.31**	**75.38**	**76.39**	15.30	30.51	39.61	46.33	52.07
	KE	WRN-16-2		73.61	**67.22**	72.30	73.23	74.00	73.79	**43.74**	**61.67**	**67.14**	**71.90**	**73.05**
	CE	None	WRN-28-2	74.83	40.58	55.76	61.37	65.45	67.74	8.92	17.35	25.28	32.72	40.45
	KD	WRN-28-4		**77.36**	54.93	65.14	68.84	71.14	72.63	9.69	19.44	27.61	35.91	44.13
	SD	WRN-28-2		76.13	59.69	68.38	71.48	**73.20**	**74.41**	14.37	34.65	45.66	52.73	58.96
	KE	WRN-16-2		73.97	**63.38**	**70.49**	**72.01**	72.91	73.55	**32.36**	**53.48**	**61.05**	**66.07**	**68.25**
CIFAR-10	CE	None	WRN-16-4	94.70	80.41	87.16	89.53	91.03	92.30	10.00	19.88	29.61	38.88	47.40
	KD	WRN-28-4		**95.55**	86.30	90.95	92.51	93.71	94.17	10.00	24.72	37.05	49.92	61.50
	SD	WRN-16-4		94.84	88.15	**92.38**	**93.50**	**94.16**	**94.30**	13.17	37.25	59.24	71.41	76.49
	KE	WRN-16-2		94.24	**88.46**	92.31	93.34	93.77	94.05	**23.96**	**57.65**	**76.53**	**81.33**	**88.14**
	CE	None	WRN-28-2	94.47	79.86	86.91	89.51	90.93	91.87	10.00	19.89	29.55	38.93	47.34
	KD	WRN-28-4		**95.13**	84.28	89.67	91.91	93.05	93.61	10.00	20.40	31.34	42.76	53.86
	SD	WRN-28-2		94.70	84.65	89.95	91.85	92.87	**93.88**	10.00	25.29	38.32	52.76	63.01
	KE	WRN-16-2		94.04	**86.90**	**91.92**	**92.95**	**93.44**	93.56	**12.97**	**45.46**	**60.98**	**70.75**	**79.75**
data set	type	teacher	student	$\zeta, \delta = 1.0$	ζ=0.02	ζ=0.04	ζ=0.06	ζ=0.08	ζ=0.1	δ=0.1	δ=0.2	δ=0.3	δ=0.4	δ=0.5
ImageNet	CE	None	RN-50	76.15	17.48	32.24	41.53	47.41	53.25	8.26	16.67	24.82	32.57	40.38
	KD	RN-152		**77.52**	30.11	51.02	59.54	64.25	67.02	37.15	53.80	61.12	65.63	68.59
	SD	RN-50		76.34	32.73	53.30	61.68	65.59	**68.13**	48.78	61.58	66.21	68.90	70.42
	KE	RN-34		73.19	**34.92**	**55.78**	**62.82**	**65.96**	67.60	**55.82**	**64.28**	**67.41**	**69.63**	**70.77**

Table 7: Comparison of BLEU scores on WMT14 En-De with different pairs of teacher and student models. In this task, since the data has no clear categories like image data, we only conduct an experiment with hyperparameter ζ. We use the same settings in Zhou et al. (2019).

type	teacher	student	ζ									
			0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
KD	AT-base	NAT-base	9.74	11.82	12.78	13.89	15.34	15.69	16.38	16.58	16.85	18.53
SD		AT-base	19.62	21.53	22.66	22.93	25.11	25.99	26.67	26.74	27.00	27.10
KE		AT-big	19.97	22.43	23.18	23.76	25.86	26.86	27.35	27.26	27.83	27.91

model trained with $\mathcal{L}_{CE}$. In addition, even among the distillation, an equal or simple teacher model facilitates student's generalization significantly better than a more in-depth and broader teacher model. In contrast, it has completely opposite results when the data used to train the teacher is entirely distilled to train a student (ζ & δ=1.0). Especially, in CIFAR-10 and CIFAR-100 when ζ=0.1 & δ=1.0, WRN-16-4 trained with KE (teacher: WRN-16-2) achieves **88.46%** and **67.22%** accuracy, being more accurate than KD (teacher: WRN-28-4), SD (teacher: WRN-16-4), and especially has **8.05%** and **23.97%** larger accuracies than CE ($\mathcal{L}_{CE}$). This consistency is also found in ImageNet (Table 6).

This observation is closely related to Xie et al. (2020); they showed that equal or larger students might be capable of treating numerous unlabeled data in terms of noise to learn through. However, even such improvement is still valid when a small amount of data can be distilled to equal or larger students. This observation contradicts the long-term belief in the KD training framework that the teacher should be cumbersome. Furthermore, we observe that KE can lead to better optima when training a student with fewer data in the case of the CIFAR-100 dataset (See the **underlines** in Table 6). This result is somewhat surprising in light of the machine learning's long-term belief that more data generalize the model better.

Based on this finding, we also test the English-to-German machine translation task using the Transformer (Vaswani et al., 2017) architecture (Table 7). Recently, for training efficiency, there have been increasing demands of building NAT models with $\mathcal{L}_{KD}$ (Zhou et al., 2019). Unlike previous approaches, we find that distilling the knowledge from an AT model to another AT model is much more efficient than to NAT model. Table 7 shows that the AT-base student model with $\zeta = 0.1$ has a 1.09 higher BLEU score than NAT-base with $\zeta = 1.0$.

Class-imbalanced sampling (OOD sampling) (ζ=1.0 columns of Table 6) We observe that a student model accurately predicts a class of data that has never been seen before (i.e., OOD data) when a student learns knowledge from a shallow and incapacious teacher. The results are similar to the results of class-balanced sampling. As Table 6 depicts, KE always outperforms CE, KD, and SD, and in particular, when ζ=1.0 & δ=0.1, KE (teacher: RN-34) has less **18.67%** and **47.56%** error than KD (teacher: RN-152) and CE in ImageNet classification.

We believe that this result may provide intuition regarding pseudo-labeling protocols (Xu et al., 2019b; Xie et al., 2020). It is generally believed that a pre-trained model with higher accuracy enables the data to be encoded into more accurate and richer information. However, as our results show, in the OOD case, it seems that the shallow and narrow model encodes the data more abundantly and transfers knowledge to other models better. Since the pseudo-labeling situation is mainly similar to the OOD, a simple model may be more suitable as an encoder.

5 Related Works

There have been debates on explaining the dark knowledge of KD. Hinton et al. (2015) first suggested that the wrong answers of a teacher rather strengthen KD via the concept of similarity information between classes. They showed that the distilled model could distinguish the data whose label does not exist in the training set. Recently, Tang et al. (2020) claimed that KD benefits from class similarity information by comparing each template (i.e., a row vector of the fully connected layer corresponding to the class). As evidence, in the distilled student, they observed high cosine similarity values between templates that share common super-class.

On the other side, Furlanello et al. (2018) asserted that a maximum value of teacher's softmax probability is similar to importance weighting by showing that permuting all of the non-argmax elements can also improve performance. Yuan et al. (2020) argued that dark knowledge serves as a label smoothing regularizer rather than as a transfer of class similarity by showing a poorly-trained or smaller teacher can boost performance. Recently, Tang et al. (2020) modified the conjecture in Furlanello et al. (2018) and showed that the sample is positively re-weighted by the prediction of the teacher's logit vector.

Some studies demonstrated that dark knowledge also releases the challenge of training a network with a fraction of the entire dataset. Kimura et al. (2018) distilled the pseudo training data generated by the teacher in an adversarial manner. Xu et al. (2019a) gained more generalization by adding unlabeled samples into the original dataset. They judged the validity of unlabeled samples by checking whether it is in the original data distribution or not. Lopes et al. (2017) showed that metadata, including the information of a pre-trained model deployment, even enables the training of a student. Recent progress has been evolving into the generation of pseudo examples to get a high accuracy under no access to the original dataset (Li et al., 2018; Nayak et al., 2019; Yoo et al., 2019). Hinton et al. (2015) observed the robustness of KD against classifying the classes of data, which is omitted from the training dataset. However, no studies still have investigated the effects of dark knowledge with respect to the number of data samples.

6 Conclusion and future research

In this work, we have revealed and summarized the behaviors of knowledge distillation (KD). First, we show that the temperature scaling in KD focuses on the logit vector matching rather than the label matching when the temperature τ grows up. In addition, we verify that an extreme logit learning model, whose KD loss is replaced with MSE, consistently outperforms any other recently modified KD methods from extensive experiments. At the example level, based on TLD, we further observe that a student even learns the teacher's degree of data uncertainty. Lastly, far interestingly, when there are a few data or a few labels among the whole dataset, the student achieves better accuracy with the incapacious teacher with a shallow depth structure than others having wider and deeper structures. We believe that this has a big impact not only on the training with temperature scaled loss but also on the classification task when there are a few data or a few labels. The design of better algorithms considering the pair of a teacher and a student is also an engaging question for future work.

REFERENCES

Angeline Aguinaldo, Ping-Yeh Chiang, Alexander Gain, Ameya Patil, Kolten Pearson, and Soheil Feizi. Compressing gans using knowledge distillation. *CoRR*, abs/1902.00159, 2019. URL http://arxiv.org/abs/1902.00159.

Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9163–9171, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Ron Banner, Itay Hubara, Elad Hoffer, and Daniel Soudry. Scalable methods for 8-bit training of neural networks. *CoRR*, abs/1805.11046, 2018. URL http://arxiv.org/abs/1805.11046.

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

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Training pruned neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

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

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor OK Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2017.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *CoRR*, abs/1706.04599, 2017a. URL http://arxiv.org/abs/1706.04599.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1321–1330. JMLR. org, 2017b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1921–1930, 2019a.

Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3779–3787, 2019b.

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

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1314–1324, 2019.

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. URL http://arxiv.org/abs/1704.04861.

Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *Advances in Neural Information Processing Systems*, pp. 2760–2769, 2018.

Taehyeon Kim, Jonghyup Kim, and Seyoung Yun. Efficient model for image classification with regularization tricks. volume 123 of *Proceedings of Machine Learning Research*, pp. 13–26, Vancouver, CA, 08–14 Dec 2020. PMLR. URL http://proceedings.mlr.press/v123/kim20a.html.

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

Akisato Kimura, Zoubin Ghahramani, Koh Takeuchi, Tomoharu Iwata, and Naonori Ueda. Few-shot learning of neural networks from scratch by pseudo example optimization. *arXiv preprint arXiv:1802.03039*, 2018.

Animesh Koratana, Daniel Kang, Peter Bailis, and Matei Zaharia. LIT: Learned intermediate representation training for model compression. volume 97 of *Proceedings of Machine Learning Research*, pp. 3509–3518, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL http://proceedings.mlr.press/v97/koratana19a.html.

Tianhong Li, Jianguo Li, Zhuang Liu, and Changshui Zhang. Few sample knowledge distillation for efficient network compression. *arXiv preprint arXiv:1812.01839*, 2018.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. In *Advances in Neural Information Processing Systems*, pp. 11669–11680, 2019.

Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

Hossein Mobahi, Mehrdad Farajtabar, and Peter L Bartlett. Self-distillation amplifies regularization in hilbert space. *arXiv preprint arXiv:2002.05715*, 2020.

Hesham Mostafa and Xin Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. *CoRR*, abs/1902.05967, 2019. URL http://arxiv.org/abs/1902.05967.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, R Venkatesh Babu, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. *arXiv preprint arXiv:1905.08114*, 2019.

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

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Suraj Srinivas and François Fleuret. Knowledge transfer with jacobian matching. *arXiv preprint arXiv:1803.00443*, 2018.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

Jiaxi Tang, Rakesh Shivanna, Zhe Zhao, Dong Lin, Anima Singh, Ed H Chi, and Sagar Jain. Understanding and improving knowledge distillation. *arXiv preprint arXiv:2002.03532*, 2020.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10687–10698, 2020.

Yixing Xu, Yunhe Wang, Hanting Chen, Kai Han, XU Chunjing, Dacheng Tao, and Chang Xu. Positive-unlabeled compression on the cloud. In *Advances in Neural Information Processing Systems*, pp. 2561–2570, 2019a.

Yixing Xu, Yunhe Wang, Hanting Chen, Kai Han, Chunjing XU, Dacheng Tao, and Chang Xu. Positive-unlabeled compression on the cloud. In *Advances in Neural Information Processing Systems 32*, pp. 2565–2574. Curran Associates, Inc., 2019b. URL `http://papers.nips.cc/paper/8525-positive-unlabeled-compression-on-the-cloud.pdf`.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4133–4141, 2017.

Jaemin Yoo, Minyong Cho, Taebum Kim, and U Kang. Knowledge extraction with no observable data. In *Advances in Neural Information Processing Systems*, pp. 2701–2710, 2019.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016a.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016b.

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 *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Chris De Sa, and Zhiru Zhang. Improving neural network quantization without retraining using outlier channel splitting. In *International Conference on Machine Learning*, pp. 7543–7552, 2019.

Chunting Zhou, Graham Neubig, and Jiatao Gu. Understanding knowledge distillation in non-autoregressive machine translation. *arXiv preprint arXiv:1911.02727*, 2019.

A APPENDIX: OTHER METHODS

We compare to the following other state-of-the-art methods from the literature:

- Fitnets: Hints for thin deep nets (Romero et al., 2014)
- Attention Transfer (AT) (Zagoruyko & Komodakis, 2016a)
- Knowledge transfer with jacobian matching (Jacobian) (Srinivas & Fleuret, 2018)
- Paraphrasing complex network: Network compression via factor transfer (FT) (Kim et al., 2018)
- Knowledge transfer via distillation of activation boundaries formed by hidden neurons (AB) (Heo et al., 2019b)
- A comprehensive overhaul of feature distillation (Overhaul) (Heo et al., 2019a)

B DETAILS OF THE SECTION 2

In this section, we discuss the derivations of such equation and theorem that we have mentioned in section 2.

B.1 EQUATION 2

$$
\begin{aligned}
\mathcal{L} &= (1-\alpha)\mathcal{L}_{CE}(p^s(\boldsymbol{x};1), q(\boldsymbol{x})) + \alpha\mathcal{L}_{KD}(p^s(\boldsymbol{x};\tau), p^t(\boldsymbol{x};\tau)) \\
&= (1-\alpha)\left[-\sum_j q_j(\boldsymbol{x}) log p_j^s(\boldsymbol{x};1)\right] + \alpha\tau^2\left[\sum_j (p_j^t(\boldsymbol{x};\tau) log p_j^t(\boldsymbol{x};\tau) - p_j^t(\boldsymbol{x};\tau) log p_j^s(\boldsymbol{x};\tau))\right] \\
\frac{\partial\mathcal{L}}{\partial \boldsymbol{z}_k^s} &= (1-\alpha)\left[-\sum_j q_j(\boldsymbol{x})\frac{\partial log p_j^s(\boldsymbol{x};1)}{\partial \boldsymbol{z}_k^s}\right] + \alpha\tau^2\left[-\sum_j p_j^t(\boldsymbol{x};\tau)\frac{\partial log p_j^s(\boldsymbol{x};\tau)}{\partial \boldsymbol{z}_k^s}\right] \\
&= (1-\alpha)\left[-\sum_j \frac{q_j(\boldsymbol{x})}{p_j^s(\boldsymbol{x};1)}\frac{\partial p_j^s(\boldsymbol{x};1)}{\partial \boldsymbol{z}_k^s}\right] + \alpha\tau^2\left[-\sum_j \frac{p_j^t(\boldsymbol{x};\tau)}{p_j^s(\boldsymbol{x};\tau)}\frac{\partial p_j^s(\boldsymbol{x};\tau)}{\partial \boldsymbol{z}_k^s}\right] \text{ where } p_j^s(\boldsymbol{x};\tau) = \frac{e^{\boldsymbol{z}_j^s/\tau}}{\sum_i e^{\boldsymbol{z}_i^s/\tau}} \\
&= (1-\alpha)\left[-\frac{q_k(\boldsymbol{x})}{p_k^s(\boldsymbol{x};1)}\frac{\partial p_k^s(\boldsymbol{x};1)}{\partial \boldsymbol{z}_k^s}\right] + \alpha\tau^2\left[-\frac{p_k^t(\boldsymbol{x};\tau)}{p_k^s(\boldsymbol{x};\tau)}\frac{\partial p_k^s(\boldsymbol{x};\tau)}{\partial \boldsymbol{z}_k^s} - \sum_{j\neq k}\frac{p_j^t(\boldsymbol{x};\tau)}{p_k^s(\boldsymbol{x};\tau)}\frac{\partial p_j^s(\boldsymbol{x};\tau)}{\partial \boldsymbol{z}_k^s}\right] \\
&\qquad\qquad (\because q_j(\boldsymbol{x}) = 0 \text{ for all } j \neq k) \\
&= (1-\alpha)(p_k^s(\boldsymbol{x};1) - q_k(\boldsymbol{x})) + \alpha\tau^2\left[\frac{1}{\tau}(p_k^s(\boldsymbol{x};\tau) - p_k^t(\boldsymbol{x};\tau))\right] \\
&\qquad\qquad \left(\because \frac{\partial p_j^s(\boldsymbol{x};\tau)}{\partial \boldsymbol{z}_k^s} = \partial\left(\frac{e^{\boldsymbol{z}_j^s/\tau}}{\sum_i e^{\boldsymbol{z}_i^s/\tau}}\right)/\partial \boldsymbol{z}_k^s = \begin{cases} \frac{1}{\tau}p_k^s(\boldsymbol{x};\tau)(1-p_k^s(\boldsymbol{x};\tau)), & \text{if } j = k \\ -\frac{1}{\tau}p_j^s(\boldsymbol{x};\tau)p_k^s(\boldsymbol{x};\tau), & \text{otherwise} \end{cases}\right) \qquad (6)
\end{aligned}
$$

B.2 PROOF OF THEOREM 1

$$\begin{aligned}
\lim_{\tau\to\infty} \tau(p_k^s(\boldsymbol{x};\tau) - p_k^t(\boldsymbol{x};\tau)) &= \lim_{\tau\to\infty} \tau \left(\frac{e^{\boldsymbol{z}_k^s/\tau}}{\sum_{j=1}^K e^{\boldsymbol{z}_j^s/\tau}} - \frac{e^{\boldsymbol{z}_k^t/\tau}}{\sum_{j=1}^K e^{\boldsymbol{z}_j^t/\tau}} \right) \\
&= \lim_{\tau\to\infty} \tau \left(\frac{1}{\sum_{j=1}^K e^{(\boldsymbol{z}_j^s - \boldsymbol{z}_k^s)/\tau}} - \frac{1}{\sum_{j=1}^K e^{(\boldsymbol{z}_j^t - \boldsymbol{z}_k^t)/\tau}} \right) \\
&= \lim_{\tau\to\infty} \tau \left(\frac{\sum_{j=1}^K e^{(\boldsymbol{z}_j^t - \boldsymbol{z}_k^t)/\tau} - \sum_{j=1}^K e^{(\boldsymbol{z}_j^s - \boldsymbol{z}_k^s)/\tau}}{\left(\sum_{j=1}^K e^{(\boldsymbol{z}_j^s - \boldsymbol{z}_k^s)/\tau}\right)\left(\sum_{j=1}^K e^{(\boldsymbol{z}_j^t - \boldsymbol{z}_k^t)/\tau}\right)} \right) \\
&= \lim_{\tau\to\infty} \left(\frac{\sum_{j=1}^K \tau\left(e^{(\boldsymbol{z}_j^t - \boldsymbol{z}_k^t)/\tau} - 1\right) - \sum_{j=1}^K \tau\left(e^{(\boldsymbol{z}_j^s - \boldsymbol{z}_k^s)/\tau} - 1\right)}{\left(\sum_{j=1}^K e^{(\boldsymbol{z}_j^s - \boldsymbol{z}_k^s)/\tau}\right)\left(\sum_{j=1}^K e^{(\boldsymbol{z}_j^t - \boldsymbol{z}_k^t)/\tau}\right)} \right) \\
&= \frac{1}{K^2} \sum_{j=1}^K \left((\boldsymbol{z}_k^s - \boldsymbol{z}_j^s) - (\boldsymbol{z}_k^t - \boldsymbol{z}_j^t) \right) \\
&\qquad \left(\because \lim_{\tau\to\infty} e^{(\boldsymbol{z}_j^f - \boldsymbol{z}_k^f)/\tau} = 1, \lim_{\tau\to\infty} \tau\left(e^{(\boldsymbol{z}_j^f - \boldsymbol{z}_k^f)/\tau} - 1\right) = \boldsymbol{z}_j^f - \boldsymbol{z}_k^f \right)
\end{aligned} \tag{7}$$

$$\lim_{\tau\to 0} = \tau(p_k^s(\boldsymbol{x};\tau) - p_k^t(\boldsymbol{x};\tau)) = 0 \;\; (\because -1 \leq p_k^s(\boldsymbol{x};\tau) - p_k^t(\boldsymbol{x};\tau) \leq 1) \tag{8}$$

B.3 DETAILED VALUES OF FIGURE 1

Table 8 and Table 9 show the detailed values in Figure 1.

alpha	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\tau = 1$	99.53	99.54	99.55	99.56	99.54	99.56	99.53	99.50	99.46	99.34
$\tau = 3$	99.37	99.09	98.69	98.33	97.85	97.43	96.84	96.26	95.76	95.05
$\tau = 5$	99.32	99.07	98.70	98.19	97.66	96.96	96.18	95.11	93.91	92.63
$\tau = 20$	99.33	99.13	98.96	98.63	98.25	97.87	97.29	96.33	95.12	92.76
$\tau = \infty$	99.35	99.22	99.02	98.80	98.42	98.11	97.60	96.49	95.42	92.74

Table 8: Training accuracy on CIFAR-100 (Teacher: WRN-28-4 & Student: WRN-16-2).

alpha	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\tau = 1$	72.79	72.56	72.80	72.70	72.84	72.68	72.78	72.60	72.87	72.90
$\tau = 3$	73.76	73.90	73.88	74.30	74.18	74.64	74.78	74.32	74.35	74.24
$\tau = 5$	73.84	74.00	74.36	74.54	74.74	74.54	75.17	74.84	75.24	74.88
$\tau = 20$	73.51	73.94	74.34	74.54	74.64	74.86	75.05	74.86	75.26	75.76
$\tau = \infty$	73.18	73.65	74.04	74.28	74.45	75.03	75.04	74.67	75.37	75.51

Table 9: Testing accuracy on CIFAR-100 (Teacher: WRN-28-4 & Student: WRN-16-2).

C RELATED WORKS

C.1 SELF-DISTILLATION (SD)

There have been increasing demands for improving the performance of SD with variety.

Distilling the information in the latent vector of teacher. In Yim et al. (2017), two identical deep neural networks are set as a teacher and a student, respectively, to facilitate the teacher's generalization. They penalized the L2-loss from each paired layer of a student and a teacher. Zhang et al. (2019) utilized auxiliary classifiers which are attached to additional bottlenecks and fully connected layers. They expected to encourage discrimination in lower stages. Experimental results depicted that this modified SD outperforms any other distillation methods. Ahn et al. (2019) proposed another approach that maximizes mutual information between teacher and student networks.

Understanding the effects of SD iterations. Furlanello et al. (2018) improved the teacher's generalization in iterative manner. Empirically, they denoted remarkable results into four folds:

1. Only using KD without ground-truth label outperforms using both the teacher's prediction and ground-truth labels.
2. KD has a similar effect to importance weighting by considering the teacher's softmax output.
3. A student distilled with teacher's output permuted except the argmax value still brings the accuracy similar to that with original teacher's output. It implies that the success of KD owes to factors other than information contained in the non-argmax output of the teacher.
4. In the KD, ensemble of iterative models also outperforms than that a single model.

Mobahi et al. (2020) provided theoretical analysis of SD in the circumstance where models are in Hilbert space and fitting these models is subject to l_2 regularization in those specific function space. Specifically, the effect of SD acts like a regularizer by restricting the number of basis functions which are used to represent the desired solution. In this setting, they asserted that SD iterations improve the model performance until a certain step and there exists a lower bound on the number of distillation iterations.

C.2 KD ON NEURAL MACHINE TRANSLATION

Neural machine translation (NMT) has shown remarkable performance in machine translation tasks. Promising NMT models consists of encoder-decoder architecture (Bahdanau et al., 2014), built up with a module which automatically search the most similar representation of the target word among source words (Vaswani et al., 2017). Certain model family translates the next target word by using the words before as inputs in an autoregressive (AT) manner. Due to this characteristic, the bottleneck always exists in the decoding step when inferencing.

There were approaches to decode in with non-autoregressive (NAT) methods by predicting the target word just by looking at each source word. This type of methods helps parallelism in inference because it doesn't need the former predicted word as a input reducing the inference time. However, NAT in machine translation is a very difficult task because there our diverse candidate for the true target word (i.e., multi modality problem). To mitigate this issue, the true target labels of the corpus have been replaced with labels from a pre-trained AT model (Kim & Rush, 2016). With this concept, NAT has shown a significant improvements (Gu et al., 2017) . Zhou et al. (2019) showed that knowledge distillation reduce the complexity in a data sets which helps NAT to deal with multi modality problem.

D EXPERIMENTAL SETTINGS

D.1 MODELS

ResNet (RN). We use the benchmark network as ResNet (He et al., 2016a), which is a champion of the ILSVRC2015. This network uses the concept of residual learning which makes layers to learn the residual between underlying mapping and input of layer. In our experiment, we compose the ResNet with a Basickblock that consists of two consecutive 3×3 convolutional layers and for each convolutional layer batch normalizaion and ReLU activation follows sequentially.

We keep almost the same settings as He et al. (2016a), but one different thing from He et al. (2016a) is that we also control the width of networks (i.e., the number of features in each convolutional layer) to systematically dissect the effects coming from the structural factors. To control the width of network, we introduce additional widening factor k. Widening factor k multiplies the number of channels in each convolutional layer thus leading to wider network.

In our experiment, we use notation RN-n-k with n total number of layers and widening factor k. Several blocks compose a group and convolutional layers in each group share the same number of channels. More specifically, RN-20-1 is network with 20 layers which is exactly the same as He et al. (2016a) and RN-20-4 is wider network with widening factor $k = 4$ that extracts more features

in each convolutional layer. In the process of distilling knowledge, other network hyperparmeters are fixed except α and τ.

Wide-ResNet (WRN). For the variety, we also use the Wide-ResNet (Zagoruyko & Komodakis, 2016b). The authors suggested to increase the width of convolutional layer not the depth of network. They added additional dropout for regularization effect for each residual block. In WRN, only basicblock from RN is used since, bottleneck block makes network thinner and WRN doesn't have interest in deepening the network. The sequence of convolution (Conv), batch normalization (BN) and ReLU activation follows BN-ReLU-Conv as He et al. (2016b). n and k denote the total number of convolutional layers n and widening factor k in WRN, respectively.

In our experiment, we explore various Wide-ResNet structures. For all WRN, each residual block contains two 3×3 convolutional layers which is a baseline in orginal paper Zagoruyko & Komodakis (2016b) with the best test accuracy in their experiments. We do not utilize dropout in our experiments. In all settings of over distillation, self distillation and under distillation, teacher-student pair is derived with different hyperparameter values of number of convolutional layers and widening factor, n, k respectively. In the process of distillation, we only control the hyperparameters α and τ to maintain consistency.

Autoregressive transformer (AT). In neural machine translation, input and output are sequence of words. Nearby words in a sequence affect each other. Autoregressive transformer (AT) models capture this phenomena by using the previous predicted word as input to predict the current word (Vaswani et al., 2017). It is widely known that AT models can mitigate the issue of high computational burdens in training through parallelism while recurrent models such as recurrent neural networks, long short-term memory (Hochreiter & Schmidhuber, 1997) and gated recurrent (Chung et al., 2014) are not.

The auto regressive model used in our experiments are applied based on transformer model. We keep the same settings of building AT models in Zhou et al. (2019) (Table 10).

Non-autoregressive transformer (NAT). For non-autoregressive model (NAT), we have used slightly shifted version of vanilla NAT model (Gu et al., 2017) whose official implementation is in Fairseq [1]. The overall architecture of original vanilla NAT is nearly identical as the Transformer except it additionally predicts fertility in the encoding step and it generates the output in non-autoregressive manner in the decoding step. However, in our paper, we don't utilize the fertility value, but simply copy the encode embedding to the decoder. We keep the same settings of building NAT models in Zhou et al. (2019)(Table 10)

Models	NAT-base	AT-base	AT-big
d_{model}	512	512	1024
d_{hidden}	2048	2048	4096
n_{layer}	6	6	6
n_{heads}	8	8	16
$p_{dropout}$	0.3	0.3	0.3

Table 10: Hyperparameters of AT, NAT models. We utilize the same notation from Vaswani *et al.* Vaswani et al. (2017). d_{model} indicates the dimension of key, value and query dimension. d_{hidden} stands for the hidden dimension of feed forward network inside the sub-layer. n_{layer}, n_{heads}, $p_{dropout}$ indicate number of encoder/decoder layers and multi-head attention module and probability of dropout respectively.

D.2 Training setups

ResNet. We run 200 epochs for each model with optimizer SGD with momentum 0.9, initial learning rate $\gamma = 0.1$ and weight decay 1×10^{-4}. Also, we applied step decay for learning rate in 100 and 150 epoch by 0.1. On the other side, scale and shift parameters β and γ in BN were trained with momentum 0.99 and without weight decay. We use the same data augmentation policies in Szegedy et al. (2016).

[1] https://github.com/pytorch/fairseq

Table 11: Ablation performance on various regularizations on CIFAR-100 dataset. 'RN' indicates the a family of ResNet. All teacher models are trained with such regularizations on the dataset whose (δ, ζ) is equal to $(1.0, 1.0)$.

teacher	student	regularization	$\zeta, \delta = 1.0$	$\delta = 1.0$					$\zeta = 1.0$				
				ζ=0.1	ζ=0.2	ζ=0.3	ζ=0.4	ζ=0.5	δ=0.1	δ=0.2	δ=0.3	δ=0.4	δ=0.5
None	RN-20-4	WD&BN&SC	76.89	41.10	56.54	63.69	67.25	69.93	8.99	17.35	25.51	33.15	41.03
		W/O WD	72.67	36.05	50.14	58.11	63.28	66.17	8.74	16.98	24.46	32.12	39.60
		W/O BN	76.15	41.21	56.73	63.52	67.07	70.07	8.97	17.49	25.63	33.20	41.17
		W/O SC	74.61	37.18	53.23	61.67	64.96	67.27	8.97	17.10	25.08	32.61	40.65
RN-20-1	RN-20-4	WD&BN&SC	69.98	66.57	69.93	71.10	70.74	70.66	52.77	65.86	68.86	69.81	69.93
		W/O WD	69.49	66.99	70.02	71.39	70.57	70.58	53.25	65.38	68.73	69.73	69.98
		W/O BN	69.72	66.27	69.96	71.31	70.55	70.50	52.99	66.01	68.82	70.02	69.82
		W/O SC	70.41	63.72	69.10	70.56	70.82	71.02	40.88	62.94	66.68	68.60	69.93

Table 12: Top1 accuracy of HCL on CIFAR-100 dataset. We keep the student the same as WRN-16-2 and use the same settings described in Nayak et al. (2019).

teacher	HCL	$\zeta, \delta = 1.0$	$\delta = 1.0$					$\zeta = 1.0$				
			ζ=0.1	ζ=0.2	ζ=0.3	ζ=0.4	ζ=0.5	δ=0.1	δ=0.2	δ=0.3	δ=0.4	δ=0.5
WRN-16-6	O	73.04	42.05	53.27	60.05	63.92	66.92	9.16	17.48	24.99	32.51	40.20
WRN-16-2	O	72.78	40.52	52.86	58.86	62.59	68.08	9.13	17.34	25.09	31.92	39.73

Wide-ResNet. We explored various types of WRN-n-k in experiment with different hyperparameter values of n and k. Data augmentation method follows from Szegedy et al. (2016) and the same training setup was used in WRN as RN. We run 200 epochs using SGD with momentum 0.9, initial learning rate $\gamma = 0.1$, dropping 0.1 in 100 and 150 epoch, weigh decay 1×10^{-4} and parameters for batch normalization momentum with 0.99 without weight decay. After exploring various values of hyperparameters α and τ, we set those values as 1.0 and 20 respectively to reveal the benefit from the number of training data in knowledge distillation.

Autoregressive transformer (AT). We have almost kept identical configuration with Tang et al. (2020) to train auto regressive transformer Vaswani et al. (2017). During Training, we run 70 epochs for each model and used Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$, $\epsilon = 1e-8$. We adopt inverse square root scheduler with 4000 warm up updates and maximum learning rate 0.0005. We utilize the label smoothing as 0.1 and treated the last model as our best model. We decode AT model using beam-search with size 5.

Non-autoregressive transformer (NAT). We train vanilla non auto regressive transformer Gu et al. (2017) 70 epochs with same optimizer setting with AT models. Also adopt inverse square scheduler, but with 10000 warm up updates with label smoothing 0.1. We treat the last model trained 90 epochs as our best model. We have not use any advanced decoding technique such as beam search, but use greedy decoding.

E KD AND OTHER FACTORS

E.1 ABLATION STUDY ON OTHER REGULARIZATIONS

We conduct an extensive ablation study by systematically eliminating three regularizations to analyze the robustness of KD towards such regularizations when a small amount of data is distilled: (1) weight decay (WD), (2) batch normalization (BN), and (3) shortcut (SC). In the setting for BN, we fix the values of both β and γ (i.e., affine parameters) in BN layers of a student model instead of elimination since the network can not be trained without BN layer. For the investigation of SC, we eliminate the shortcut of a student model, including identity mapping. The results in Table 11 show that, interestingly, even without such regularizations, a student model trained with KD has a similar characteristic, which fewer data leads to better generalization. Additionally, other consistent tendencies described in section 4 also happen.

E.2 HAND-CRAFTED LABEL OF CLASS SIMILARITY

We study how the prior in the form of class similarities (Nayak et al., 2019; Kim et al., 2020) affects the benefit from the number of training data compared to KD. Specifically, we experiment with the use of a hand-crafted label (HCL) that utilizes the prior knowledge of class similarities from the teacher (Nayak et al., 2019) (Table 12). Artificially creating a soft target with hand-craft by importing class-wise information from the teacher is not valid when there are few data, while the performance slightly increases when $\delta, \zeta = 1.0$.

In details of constructing hand-crafted labels, we first obtain the class similarity matrix from the teacher. The class similarity is calculated as follows:

$$C(i,j) = \frac{w_i^T w_j}{||w_i||||w_j||}$$

where $C(i,j)$ is the (i,j) elements in class similarity matrix C, and w_i is the i-th row vector of fully-connected layer's weight matrix.

Crafting labels via Dirichlet sampling (Nayak et al., 2019; Kim et al., 2020) Nayak et al. (2020) and Kim et al. (2020) proposed a method of crafting the labels via Dirichlet distribution whose the concentration parameter α is considered as the row vector α_k corresponding to class k. Here, they handle the amount of distillation with a hyperparameter β:

$$p(s) = Dir(K, \beta \times \alpha)$$

where K is the number of classes, and β is a scaling factor. In our work, we conduct an experiment with the class similarity matrix of WRN-28-4 and $\beta = 1.0$.

E.3 CALIBRATION

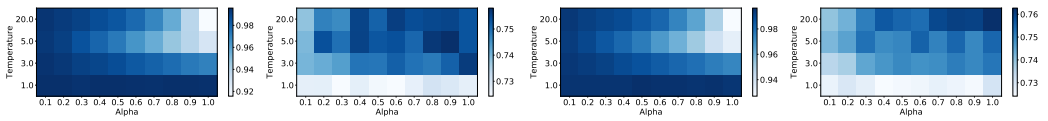
In this subsection, we evaluate the calibration effects of KD as τ increases. To measure calibration, we use the estimated expected calibration error (ECE), which is a quantitative metric of calibration Guo et al. (2017b); Naeini et al. (2015), in Table 13. The results demonstrate that a student with large τ attempts to follow the logit distribution of the teacher.

Table 13: ECE of the training samples. Here, (student, teacher) is (WRN-28-4, WRN-16-2), and all student models are trained with $\alpha = 1.0$.

	teacher	CE	KD (τ=1)	KD (τ=3)	KD (τ=5)	KD (τ=20)
ECE	0.86	3.40	4.64	0.61	0.63	0.78

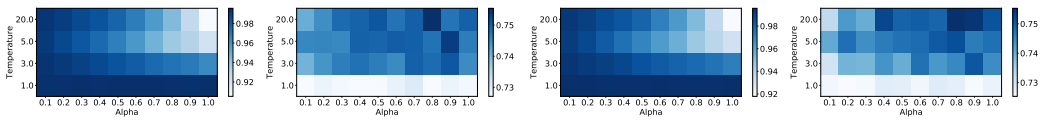
E.4 VARIOUS PAIRS OF TEACHERS AND STUDENTS

We provide the results that support the Figure 1 in various pairs of teachers and students (Figure 4). All figures depict that higher the value of α, smaller is the over-fitting problem (i.e., low training accuracy, but high test accuracy) under the condition that τ is 20.



(a) T: WRN-16-4 & S: WRN-16-2

(b) T: WRN-16-6 & S: WRN-16-2



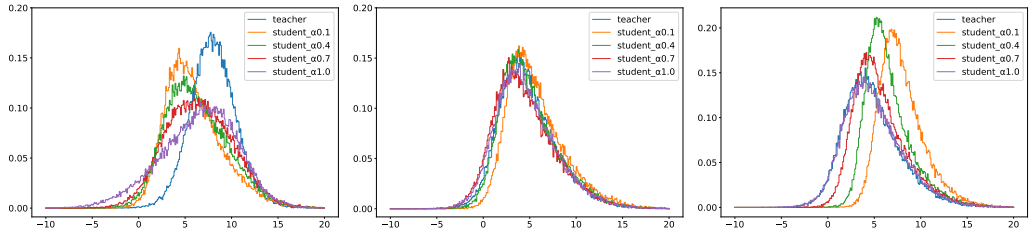
(c) T: WRN-28-2 & S: WRN-16-2

(d) T: WRN-40-2 & S: WRN-16-2

Figure 4: **Grip maps of accuracies according to the change of α and τ** on CIFAR-100 when (a) (teacher, student) = (WRN-16-4, WRN-16-2), (b) (teacher, student) = (WRN-16-6, WRN-16-2), (c) (teacher, student) = (WRN-28-2, WRN-16-2), and (d) (teacher, student) = (WRN-40-2, WRN-16-2). The left grid maps presents training top1 accuracies, and the right grid maps presents test top1 accuracies.

E.5 THE BALANCING HYPERPARAMETER α IN $\mathcal{L}$

Here, we further investigate the regularization effects of α based on TLD. Increasing α has similar results compared to that of τ (Figure 5). In the case of UD and SD, a student learns the teacher's logit distribution almost fully when α gets closer to 1.0. On the other side, when the student is more straightforward than the teacher, the student is not able to learn the exact teacher's logit distribution. As we've already mentioned in section 2, it is due to the bottleneck coming from the student's low complexity.



(a) WRN-28-4 to WRN-16-2. (b) WRN-16-2 to WRN-16-2. (c) WRN-16-2 to WRN-28-4.

Figure 5: Pdf of TLD. All students are trained with τ=20, and all teachers are trained with CE. Both models share other training recipes such as learning rate, batch size, and weight decay.

E.6 EXAMPLE RE-WEIGHTING (TANG ET AL., 2020)

In this subsection, we evaluate the example re-weighting of KD based on teacher model's prediction confidence on the ground-truth class (Tang et al., 2020). Refer to Tang et al. (2020), we raise a question of whether the relationship between a re-weighting value of a sample (i.e., $\tau \left(\frac{p^t(\tau)_k - p^s(\tau)_k}{1 - q^s(1)_k} \right)$) and the confidence of teacher on the sample keeps positive during in the course of training (Figure 6). Here, x-axis means the softened softmax value of teacher for ground-truth class, i.e., $p^t(\tau)_k$, and y-axis means the log value of re-weight factor when $\alpha = 1$, i.e., $\tau \left(\frac{p^t(\tau)_k - p^s(\tau)_k}{1 - q^s(1)_k} \right)$.

Figure 6 shows the result of the relationship between example re-weighting and teacher's predicted label according to the changes of training iterations until the second epoch begins. Similar to the results of Tang et al. (2020), we also find that there is a positive correlation between the effect or example re-weighting and the teacher's softened top1 prediction in the early stage of training. Especially, this correlation seems to be strengthened (Figure 6). On the other side, after 1 epoch, this trend continues and no significant changes happen in the epoch of learning rate decay.

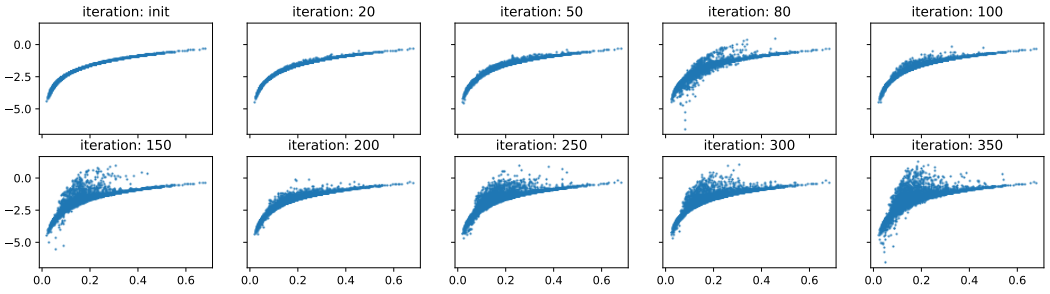


Figure 6: Re-weighting factor scatterplot in the first epoch.

E.7 REGULARIZATION OF THE LARGE LEARNING RATE

Recently, Li et al. (2019) demonstrated that a large learning rate model with annealing generalizes better on hard-to-generalize and easier-to-fit patterns than its small learning rate. Here, we aim to study whether KD facilitates the regularization effect of this concept. Through the findings in section 3, we consider the data that has hard-to-generalize and easier-to-fit patterns as high TLD

valued data from teacher. For the experiment, we train models for 200 epochs with an initial learning rate of 0.1, and the learning rate is annealed when the epoch is 100 and 150.

Table 14: Top1 training accuracy and the number of samples whose TLD value from student is above the mean TLD of teacher (i.e., mean = 7.99), when the learning rate is annealed. Here, (teacher, student) is (WRN-28-4, WRN-16-2). α in $\mathcal{L}$ is set to 1.0.

Learning rate	CE	KD			
		τ=1	τ=20	τ=∞	MSE
0.1	64.73 (2655)	65.25 (2445)	67.87 (7282)	70.75 (7383)	67.59 (7157)
0.01	92.08 (9440)	92.48 (8291)	86.62 (16006)	86.40 (14027)	87.32 (14989)
0.001	99.48 (13269)	99.45 (10573)	92.89 (18983)	92.74 (17374)	92.70 (17440)

Table 14 shows that $\tau = 20$ or ∞ and MSE correctly predicts far more numbers of hard-to-generalized and easy-to-fitted data than others (CE and $\tau = 1$), especially in the phase of the initial learning rate. We believe that, as we mentioned in theory, this difference is from whether model strives to be logit learning or label learning. Furthermore, this effect still works in annealed learning rates while the discrepancy slightly is reduced.

E.8 Details of the section 4

In this subsection, we discuss the results that we do not show in section 4 (Table 15).

Table 15: Comparison of top1 accuracies on CIFAR-100 with a family of ResNets. All teacher models are trained with the dataset whose (δ, ζ) is equal to $(1.0, 1.0)$. We report the best result over 3 individual runs with different initializations.

	teacher	student	$\zeta, \delta = 1.0$	$\delta = 1.0$					$\zeta = 1.0$				
				ζ=0.1	ζ=0.2	ζ=0.3	ζ=0.4	ζ=0.5	δ=0.1	δ=0.2	δ=0.3	δ=0.4	δ=0.5
CE	None	RN-20-1	72.56	42.51	54.76	60.15	63.74	66.65	9.13	17.06	24.99	32.33	39.55
		RN-20-4	76.89	41.10	56.54	63.69	67.25	69.93	8.99	17.35	25.51	33.15	41.03
KE	RN-20-1	RN-20-2	69.57	63.85	68.51	69.91	70.25	69.96	44.15	61.21	65.46	68.93	69.4
		RN-20-3	69.48	65.46	69.12	70.49	70.64	70.35	49.98	64.73	67.99	69.92	69.60
		RN-20-4	69.98	66.57	69.93	71.10	70.74	70.66	52.77	65.86	68.86	69.81	69.93
		RN-50-1	69.63	59.66	66.87	68.82	69.51	70.34	36.18	55.23	61.84	65.15	67.44
		RN-110-1	70.98	60.09	67.78	69.61	70.13	70.73	32.83	53.08	64.18	67.25	68.21
		RN-152-1	71.12	58.49	68.39	69.81	70.53	70.53	30.07	57.96	64.72	67.30	68.52