Comparing Kullback-Leibler Divergence and Mean Squared Error Loss in Knowledge Distillation

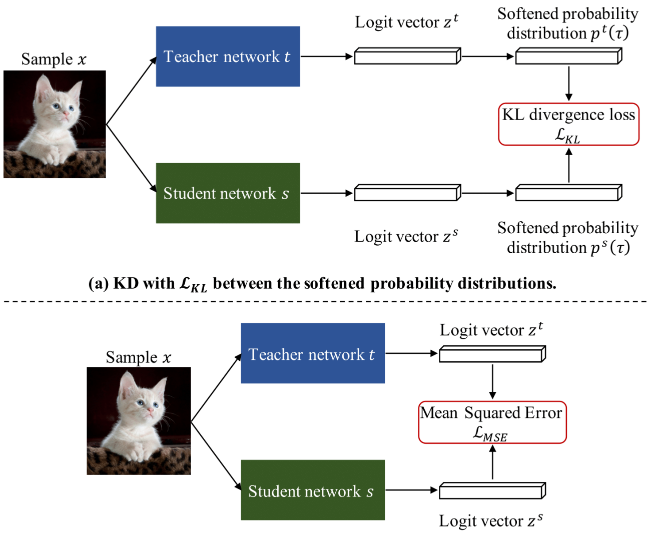
Taehyeon Kim[*1], **Jaehoon Oh**[*2], **Nak Yil Kim**[1], **Sangwook Cho**[1] and **Se-Young Yun**[1]
[1]Graduate School of Artificial Intelligence, KAIST
[2]Graduate School of Knowledge Service Engineering, KAIST
{potter32, jhoon.oh, nakyilkim, sangwookcho, yunseyoung}@kaist.ac.kr

Abstract

Knowledge distillation (KD), transferring knowledge from a cumbersome teacher model to a lightweight student model, has been investigated to design efficient neural architectures. Generally, the objective function of KD is the Kullback-Leibler (KL) divergence loss between the softened probability distributions of the teacher model and the student model with the temperature scaling hyperparameter τ. Despite its widespread use, few studies have discussed the influence of such softening on generalization. Here, we theoretically show that the KL divergence loss focuses on the *logit matching* when τ increases and the *label matching* when τ goes to 0 and empirically show that the logit matching is positively correlated to performance improvement in general. From this observation, we consider an intuitive KD loss function, the mean squared error (MSE) between the logit vectors, so that the student model can directly learn the logit of the teacher model. The MSE loss outperforms the KL divergence loss, explained by the difference in the penultimate layer representations between the two losses. Furthermore, we show that sequential distillation can improve performance and that KD, particularly when using the KL divergence loss with small τ, mitigates the label noise. The code to reproduce the experiments is publicly available online at https://github.com/jhoon-oh/kd_data/.

1 Introduction

Knowledge distillation (KD) is one of the most potent model compression techniques in which knowledge is transferred from a cumbersome model (teacher) to a single small model (student) [Hinton *et al.*, 2015]. In general, the objective of training a smaller student network in the KD framework is formed as a linear summation of two losses: cross-entropy (CE) loss with "hard" targets, which are one-hot ground-truth vectors of the samples, and Kullback-Leibler (KL) divergence loss with the teacher's predictions. Specifically, KL divergence loss has achieved considerable success by controlling the softness of "soft" targets via the temperature-scaling hyperparameter τ. Utilizing a larger value for this hyperparameter τ makes the softmax vectors smooth over classes. Such a re-scaled output probability vector by τ is called the softened probability distribution, or the softened softmax [Maddison *et al.*, 2016; Jang *et al.*, 2016]. Recent KD has evolved to give more importance to the KL divergence loss to improve performance when balancing the objective between CE loss and KL divergence loss [Hinton *et al.*, 2015; Tian *et al.*, 2019]. Hence, we focus on training a student network based solely on the KL divergence loss.


Sample x
Teacher network t
Logit vector z^t
Softened probability distribution $p^t(\tau)$
KL divergence loss $\mathcal{L}_{KL}$
Student network s
Logit vector z^s
Softened probability distribution $p^s(\tau)$

(a) KD with $\mathcal{L}_{KL}$ between the softened probability distributions.

(b) KD with $\mathcal{L}_{MSE}$ between the logit vectors.

Figure 1: Overview of knowledge distillation (KD) without the CE loss $\mathcal{L}_{CE}$ with a ground-truth vector: KD framework with (a) KL divergence loss $\mathcal{L}_{KL}$ and (b) mean squared error loss $\mathcal{L}_{MSE}$.

Recently, there has been an increasing demand for investigating the reasons for the superiority of KD. [Yuan *et al.*, 2020; Tang *et al.*, 2020] empirically showed that the facilitation of KD is attributed to not only the privileged information on similarities among classes but also the label smoothing regularization. In some cases, the theoretical reasoning for using "soft" targets is clear. For example, in deep linear

[*] The authors contributed equally to this paper.

neural networks, KD not only accelerates the training convergence but also helps in reliable training [Phuong and Lampert, 2019]. In the case of self-distillation (SD), where the teacher model and the student model are the same, such an approach progressively restricts the number of basis functions to represent the solution [Mobahi *et al.*, 2020]. However, there is still a lack of understanding of how the degree of softness affects the performance.

In this paper, we first investigate the characteristics of a student trained with KL divergence loss with various τ, both theoretically and empirically. We find that the student's logit (i.e., an input of the softened softmax function) more closely resembles the teacher's logit as τ increases, but not completely. Therefore, we design a direct logit learning scheme by replacing the KL divergence loss between the softened probability distributions of a teacher network and a student network (Figure 1(a)) with the mean squared error (MSE) loss between the student's logit and the teacher's logit (Figure 1(b)). Our contributions are summarized as follows:

- We investigate the role of the softening hyperparameter τ theoretically and empirically. A large τ, that is, strong softening, leads to *logit matching*, whereas a small τ results in training *label matching*. In general, *logit matching* has a better generalization capacity than *label matching*.
- We propose a direct logit matching scheme with the MSE loss and show that the KL divergence loss with any value of τ cannot achieve complete logit matching as much as the MSE loss. Direct training results in the best performance in our experiments.
- We theoretically show that the KL divergence loss makes the model's penultimate layer representations elongated than those of the teacher, while the MSE loss does not. We visualize the representations using the method proposed by [Müller *et al.*, 2019].
- We show that sequential distillation, the MSE loss after the KL divergence loss, can be a better strategy than direct distillation when the capacity gap between the teacher and the student is large, which contrasts [Cho and Hariharan, 2019].
- We observe that the KL divergence loss, with low τ in particular, is more efficient than the MSE loss when the data have incorrect labels (*noisy label*). In this situation, extreme logit matching provokes bad training, whereas the KL divergence loss mitigates this problem.

2 Related Work

2.1 Knowledge Distillation

KD has been extended to a wide range of methods. One attempted to distill not only the softened probabilities of the teacher network but also the hidden feature vector so that the student could be trained with rich information from the teacher [Romero *et al.*, 2014; Zagoruyko and Komodakis, 2016a; Srinivas and Fleuret, 2018; Kim *et al.*, 2018; Heo *et al.*, 2019b; Heo *et al.*, 2019a]. The KD approach 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 compressions. In the generative models, a generator can be compressed by distilling the latent features from a cumbersome generator [Aguinaldo *et al.*, 2019].

To explain the efficacy of KD, [Furlanello *et al.*, 2018] asserted that the maximum value of a teacher's softmax probability was similar to weighted importance by showing that permuting all of the non-argmax elements could also improve performance. [Yuan *et al.*, 2020] argued that "soft" targets served as a label smoothing regularizer rather than as a transfer of class similarity by showing that a poorly trained or smaller-size teacher model can boost performance. Recently, [Tang *et al.*, 2020] modified the conjecture in [Furlanello *et al.*, 2018] and showed that the sample was positively reweighted by the prediction of the teacher's logit vector.

2.2 Label Smoothing

Smoothing the label $\boldsymbol{y}$ is a common method for improving the performance of deep neural networks by preventing the overconfident predictions [Szegedy *et al.*, 2016]. Label smoothing is a technique that facilitates the generalization by replacing a ground-truth one-hot vector $\boldsymbol{y}$ with a weighted mixture of hard targets $\boldsymbol{y}^{LS}$:

$$\boldsymbol{y}_k^{LS} = \begin{cases} (1-\beta) & \text{if } \boldsymbol{y}_k = 1 \\ \frac{\beta}{K-1} & \text{otherwise} \end{cases} \tag{1}$$

where k indicates the index, and β is a constant. This implicitly ensures that the model is well-calibrated [Müller *et al.*, 2019]. Despite its improvements, [Müller *et al.*, 2019] observed that the teacher model trained with LS improved its performance, whereas it could hurt the student's performance. [Yuan *et al.*, 2020] demonstrated that KD might be a category of LS by using the adaptive noise, i.e., KD is a label regularization method.

3 Preliminaries: KD

We denote the softened probability vector with a temperature-scaling hyperparameter τ for a network f as $\boldsymbol{p}^f(\tau)$, given a sample $\boldsymbol{x}$. The k-th value of the softened probability vector $\boldsymbol{p}^f(\tau)$ is denoted by $\boldsymbol{p}_k^f(\tau) = \frac{\exp(\boldsymbol{z}_k^f/\tau)}{\sum_{j=1}^K \exp(\boldsymbol{z}_j^f/\tau)}$, where $\boldsymbol{z}_k^f$ is the k-th value of the logit vector $\boldsymbol{z}^f$, K is the number of classes, and $\exp(\cdot)$ is the natural exponential function. Then, given a sample $\boldsymbol{x}$, the typical loss $\mathcal{L}$ for a student network is a linear combination of the cross-entropy loss $\mathcal{L}_{CE}$ and the Kullback-Leibler divergence loss $\mathcal{L}_{KL}$:

$$\begin{gathered} \mathcal{L} = (1-\alpha)\mathcal{L}_{CE}(\boldsymbol{p}^s(1), \boldsymbol{y}) + \alpha\mathcal{L}_{KL}(\boldsymbol{p}^s(\tau), \boldsymbol{p}^t(\tau)), \\ \mathcal{L}_{CE}(\boldsymbol{p}^s(1), \boldsymbol{y}) = \sum_j -\boldsymbol{y}_j \log \boldsymbol{p}_j^s(1) \\ \mathcal{L}_{KL}(\boldsymbol{p}^s(\tau), \boldsymbol{p}^t(\tau)) = \tau^2 \sum_j \boldsymbol{p}_j^t(\tau) \log \frac{\boldsymbol{p}_j^t(\tau)}{\boldsymbol{p}_j^s(\tau)} \end{gathered} \tag{2}$$

where s indicates the student network, t indicates the teacher network, $\boldsymbol{y}$ is a one-hot label vector of a sample $\boldsymbol{x}$, and α is a

Notation	Description
$\boldsymbol{x}$	Sample
$\boldsymbol{y}$	Ground-truth one-hot vector
K	Number of classes in the dataset
f	Neural network
$\boldsymbol{z}^f$	Logit vector of a sample $\boldsymbol{x}$ through a network f
$\boldsymbol{z}_k^f$	Logit value corresponding the k-th class label, i.e., the k-th value of $z^f(\boldsymbol{x})$
α	Hyperparameter of the linear combination
τ	Temperature-scaling hyperparameter
$\boldsymbol{p}^f(\tau)$	Softened probability distribution with τ of a sample $\boldsymbol{x}$ for a network f
$\boldsymbol{p}_k^f(\tau)$	The k-th value of a softened probability distribution, i.e., $\frac{exp(\boldsymbol{z}_k^f/\tau)}{\sum_{j=1}^K exp(\boldsymbol{z}_j^f/\tau)}$
$\mathcal{L}_{CE}$	Cross-entropy loss
$\mathcal{L}_{KL}$	Kullback-Leibler divergence loss
$\mathcal{L}_{MSE}$	Mean squared error loss

Table 1: Mathematical terms and notations in our work.

hyperparameter of the linear combination. For simplicity of notation, $\mathcal{L}_{CE}(\boldsymbol{p}^s(1), \boldsymbol{y})$ and $\mathcal{L}_{KL}(\boldsymbol{p}^s(\tau), \boldsymbol{p}^t(\tau))$ are denoted by $\mathcal{L}_{CE}$ and $\mathcal{L}_{KL}$, respectively. The standard choices are $\alpha = 0.1$ and $\tau \in \{3, 4, 5\}$ [Hinton *et al.*, 2015; Zagoruyko and Komodakis, 2016a].

In [Hinton *et al.*, 2015], given a single sample $\boldsymbol{x}$, the gradient of $\mathcal{L}_{KL}$ with respect to $\boldsymbol{z}_k^s$ is as follows:

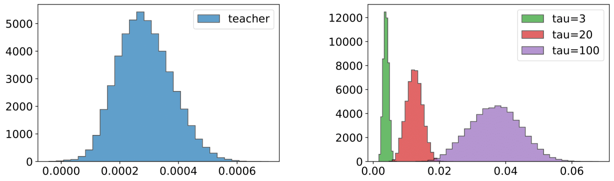
$$\frac{\partial \mathcal{L}_{KL}}{\partial \boldsymbol{z}_k^s} = \tau(\boldsymbol{p}_k^s(\tau) - \boldsymbol{p}_k^t(\tau)) \tag{3}$$

When τ goes to ∞, this gradient is simplified with the approximation, i.e., $exp(\boldsymbol{z}_k^f/\tau) \approx 1 + \boldsymbol{z}_k^f/\tau$:

$$\frac{\partial \mathcal{L}_{KL}}{\partial \boldsymbol{z}_k^s} \approx \tau \left(\frac{1 + \boldsymbol{z}_k^s/\tau}{K + \sum_j \boldsymbol{z}_j^s/\tau} - \frac{1 + \boldsymbol{z}_k^t/\tau}{K + \sum_j \boldsymbol{z}_j^t/\tau} \right) \tag{4}$$

Here, the authors assumed the zero-mean teacher and student logit, i.e., $\sum_j \boldsymbol{z}_j^t = 0$ and $\sum_j \boldsymbol{z}_j^s = 0$, and hence $\frac{\partial \mathcal{L}_{KL}}{\partial \boldsymbol{z}_k^s} \approx \frac{1}{K}(\boldsymbol{z}_k^s - \boldsymbol{z}_k^t)$. This indicates that minimizing $\mathcal{L}_{KL}$ is equivalent to minimizing the mean squared error $\mathcal{L}_{MSE}$, that is, $||\boldsymbol{z}^s - \boldsymbol{z}^t||_2^2$, under a sufficiently large temperature τ and the zero-mean logit assumption for both the teacher and the student.

However, we observe that this assumption does not seem appropriate and hinders complete understanding by ignoring the hidden term in $\mathcal{L}_{KL}$ when τ increases. Figure 2 describes the histograms for the magnitude of logit summations on the training dataset. The logit summation histogram from the teacher network trained with $\mathcal{L}_{CE}$ is almost zero (Figure 2(a)), whereas that from the student network trained with $\mathcal{L}_{KL}$ using the teacher's knowledge goes far from zero as τ increases (Figure 2(b)). This is discussed in detail in Section 4.2.



(a) Teacher.

(b) Student with $\mathcal{L}_{KL}$.

Figure 2: Histograms for the magnitudes of logit summation on the CIFAR-100 training dataset. We use a (teacher, student) pair of (WRN-28-4, WRN-16-2).

3.1 Experimental Setup

In this paper, we used an experimental setup similar to that in [Heo *et al.*, 2019a; Cho and Hariharan, 2019]: image classification on CIFAR-100 with a family of Wide-ResNet (WRN) [Zagoruyko and Komodakis, 2016b] and ImageNet with a family of of ResNet (RN) [He *et al.*, 2016]. We used a standard PyTorch SGD optimizer with a momentum of 0.9, weight decay, and apply standard data augmentation. Other than those mentioned, the training settings from the original papers [Heo *et al.*, 2019a; Cho and Hariharan, 2019] were used.

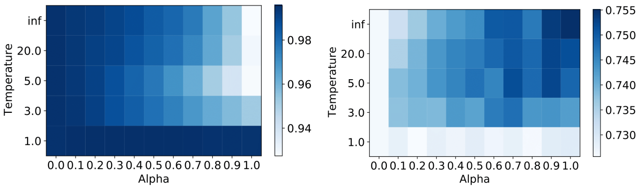
4 Relationship between $\mathcal{L}_{KL}$ and $\mathcal{L}_{MSE}$

In this section, we conduct extensive experiments and systematically break down the effects of τ in $\mathcal{L}_{KL}$ based on theoretical and empirical results. Then, we highlight the relationship between $\mathcal{L}_{KL}$ and $\mathcal{L}_{MSE}$. Then, we compare the models trained with $\mathcal{L}_{KL}$ and $\mathcal{L}_{MSE}$ in terms of performance and penultimate layer representations. Finally, we investigate the effects of a noisy teacher on the performance according to the objective.

4.1 Hyperparameter τ in $\mathcal{L}_{KL}$

We investigate the training and test accuracies according to the change in α in $\mathcal{L}$ and τ in $\mathcal{L}_{KL}$ (Figure 3). First, we empirically observe that the generalization error of a student model decreases as α in $\mathcal{L}$ increases. This means that "soft" targets are more efficient than "hard" targets in training a student if "soft" targets are extracted from a well-trained teacher. This result is consistent with prior studies that addressed the efficacy of "soft" targets [Furlanello *et al.*, 2018; Tang *et al.*, 2020]. Therefore, we focus on the situation where "soft" targets are used to train a student model solely, that is, $\alpha = 1.0$, in the remainder of this paper.

When $\alpha = 1.0$, the generalization error of the student model decreases as τ in $\mathcal{L}_{KL}$ increases. These consistent tendencies according to the two hyperparameters, α and τ, are the same across various teacher-student pairs. To explain this phenomenon, we extend the gradient analysis in Section 3 without the assumption that the mean of the logit vector is zero.



(a) Training accuracy.

(b) Test accuracy.

Figure 3: **Grid 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 top-1 accuracies** and (b) **test top-1 accuracies**. $\mathcal{L}_{KL}$ with $\tau = \infty$ is implemented using a handcrafted gradient (Eq. (5))

Proposition 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 brackets is true and 0 otherwise. Then,*

$$\begin{aligned}\lim_{\tau\to\infty}\frac{\partial\mathcal{L}_{KL}}{\partial\boldsymbol{z}_k^s} &= \frac{1}{K^2}\sum_{j=1}^{K}\left((\boldsymbol{z}_k^s-\boldsymbol{z}_j^s)-(\boldsymbol{z}_k^t-\boldsymbol{z}_j^t)\right)\\ &= \frac{1}{K}\left(\boldsymbol{z}_k^s-\boldsymbol{z}_k^t\right)-\frac{1}{K^2}\sum_{j=1}^{K}\left(\boldsymbol{z}_j^s-\boldsymbol{z}_j^t\right)\end{aligned} \tag{5}$$

$$\lim_{\tau\to 0}\frac{1}{\tau}\frac{\partial\mathcal{L}_{KL}}{\partial\boldsymbol{z}_k^s} = \mathbf{1}_{[\arg\max_j \boldsymbol{z}_j^s=k]}-\mathbf{1}_{[\arg\max_j \boldsymbol{z}_j^t=k]} \tag{6}$$

Proposition 1 explains the consistent trends as follows. In the course of regularizing $\mathcal{L}_{KL}$ with sufficiently large τ, the student model attempts to imitate the logit distribution of the teacher model. Specifically, a larger τ is linked to a larger $\mathcal{L}_{KL}$, making the logit vector of the student similar to that of the teacher (i.e., *logit matching*). Hence, "soft" targets are being fully used as τ increases. This is implemented using a handcrafted gradient (top row of Figure 3). On the other hand, when τ is close to 0, the gradient of $\mathcal{L}_{KL}$ does not consider the logit distributions and only identifies whether the student and the teacher share the same output (i.e., *label matching*), which transfers limited information. In addition, there is a scaling issue when τ approaches 0. As τ decreases, $\mathcal{L}_{KL}$ increasingly loses its quality and eventually becomes less involved in learning. The scaling problem can be easily fixed by multiplying $1/\tau$ by $\mathcal{L}_{KL}$ when τ is close to zero.

From this proposition, it is recommended to modify the original $\mathcal{L}_{KL}$ in Eq. (2), considering $\tau \in (0, \infty)$ as follows:

$$\max(\tau,\tau^2)\sum_j \boldsymbol{p}_j^t(\tau)\log\frac{\boldsymbol{p}_j^t(\tau)}{\boldsymbol{p}_j^s(\tau)} \tag{7}$$

The key difference between our analysis and the preliminary analysis on a sufficiently large τ, i.e., in Eq. (5), is that the latter term is generated by removing the existing assumption on the logit mean, which is discussed in Section 4.2 at the loss-function level.

4.2 Extensions from $\mathcal{L}_{KL}$ to $\mathcal{L}_{MSE}$

In this subsection, we focus on Eq. (5) to investigate the reason as to why the efficacy of KD is observed when τ is greater than 1 in the KD environment, as shown in Figure 3. Eq. (5) can be understood as a biased regression of the vector expression as follows:

Student	$\mathcal{L}_{CE}$	$\mathcal{L}_{KL}$					$\mathcal{L}_{MSE}$
		τ=1	τ=3	τ=5	τ=20	τ=∞	
WRN-16-2	72.68	72.90	74.24	74.88	75.15	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**

Table 2: Top-1 test accuracies on CIFAR-100. WRN-28-4 is used as a teacher for $\mathcal{L}_{KL}$ and $\mathcal{L}_{MSE}$.

$$\lim_{\tau\to\infty}\nabla_{\boldsymbol{z}^s}\mathcal{L}_{KL} = \frac{1}{K}\left(\boldsymbol{z}^s-\boldsymbol{z}^t\right)-\frac{1}{K^2}\sum_{j=1}^{K}\left(\boldsymbol{z}_j^s-\boldsymbol{z}_j^t\right)\cdot\mathbb{1}$$

where $\mathbb{1}$ is a vector whose elements are equal to one. Furthermore, we can derive the relationship between $\lim_{\tau\to\infty}\mathcal{L}_{KL}$ and $\mathcal{L}_{MSE}$ as follows:

$$\begin{aligned}\lim_{\tau\to\infty}\mathcal{L}_{KL} &= \frac{1}{2K}||\boldsymbol{z}^s-\boldsymbol{z}^t||_2^2+\delta_\infty = \frac{1}{2K}\mathcal{L}_{MSE}+\delta_\infty\\ \delta_\infty &= -\frac{1}{2K^2}(\sum_{j=1}^{K}\boldsymbol{z}_j^s-\sum_{j=1}^{K}\boldsymbol{z}_j^t)^2+Constant\end{aligned} \tag{8}$$

In Figure 2(a), the sum of the logit values of the teacher model is almost zero. With the teacher's logit value, δ_∞ is approximated as $-\frac{1}{2K^2}(\sum_{j=1}^{K}\boldsymbol{z}_j^s)^2$. Therefore, δ_∞ can make the logit mean of the student trained with $\mathcal{L}_{KL}$ depart from zero. From this analysis, it is unreasonable to assume that the student's logit mean is zero. We empirically find that the student's logit mean breaks the existing assumption as τ increases (Figure 2(b)). In summary, δ_∞ *hinders complete logit matching by shifting the mean of the elements in the logit*. In other words, as derived from Eq. (8), optimizing $\mathcal{L}_{KL}$ with sufficiently large τ is equivalent to optimizing $\mathcal{L}_{MSE}$ with the additional regularization term δ_∞, and it seems to rather hinder logit matching.

Therefore, we propose the direct logit learning objective for enhanced logit matching as follows:

$$\begin{aligned}\mathcal{L}' &= (1-\alpha)\mathcal{L}_{CE}(\boldsymbol{p}^s(1),\boldsymbol{y})+\alpha\mathcal{L}_{MSE}(\boldsymbol{z}^s,\boldsymbol{z}^t),\\ \mathcal{L}_{MSE}(\boldsymbol{z}^s,\boldsymbol{z}^t) &= ||\boldsymbol{z}^s-\boldsymbol{z}^t||_2^2\end{aligned} \tag{9}$$

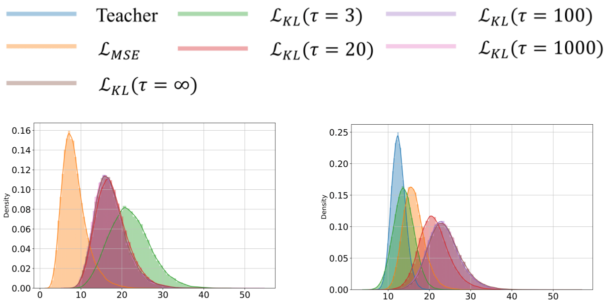
Although this direct logit learning was used in [Ba and Caruana, 2013; Urban *et al.*, 2016], they did not investigate the wide range of temperature scaling and the effects of MSE in the latent space. In this respect, our work differs.

4.3 Comparison of $\mathcal{L}_{KL}$ and $\mathcal{L}_{MSE}$

We empirically compared the objectives $\mathcal{L}_{KL}$ and $\mathcal{L}_{MSE}$ in terms of performance gains and measured the distance between the logit distributions. Following the previous analysis, we also focused on "soft" targets in $\mathcal{L}'$. Table 2 presents the top-1 test accuracies on CIFAR-100 according to the student

Student	Baseline	SKD [2015]	FitNets [2014]	AT [2016a]	Jacobian [2018]	FT [2018]	AB [2019b]	Overhaul [2019a]	MSE
WRN-16-2	72.68	73.53	73.70	73.44	73.29	74.09	73.98	**75.59**	75.54
WRN-16-4	77.28	78.31	78.15	77.93	77.82	78.28	78.64	78.20	**79.03**
WRN-28-2	75.12	76.57	76.06	76.20	76.30	76.59	76.81	76.71	**77.28**

Table 3: **Test accuracy of various KD methods** on CIFAR-100. All student models share the same teacher model as WRN-28-4. The standard KD (SKD) represents the KD method [Hinton *et al.*, 2015] with hyperparameter values (α, τ) used in Eq. (2) as (0.1, 5). MSE represents the KD with $\mathcal{L}_{MSE}$ between logits; the Overhaul [Heo *et al.*, 2019a] model is reproduced by using our pretrained teacher, and the others are the results reported in [Heo *et al.*, 2019a]. The baseline indicates the model trained with $\mathcal{L}_{CE}$ without the teacher model.



(a) $\|\boldsymbol{z}^s - \boldsymbol{z}^t\|_2$

(b) Norm of pre-logit

Figure 4: **(a)** Probabilistic density function (pdf) for $\|\boldsymbol{z}^s - \boldsymbol{z}^t\|_2$ on CIFAR-100 training dataset; **(b)** The pdf for the 2-norm of pre-logit (i.e., $\|r^s\|_2$) on CIFAR-100 training dataset. We use a (teacher, student) pair of (WRN-28-4, WRN-16-2).

learning scheme for various teacher-student pairs. The students trained with $\mathcal{L}_{CE}$ are vanilla models without a teacher. The students trained with $\mathcal{L}_{KL}$ or $\mathcal{L}_{MSE}$ are trained following the KD framework without using the "hard" targets, i.e., $\alpha = 1$ in $\mathcal{L}$ and $\mathcal{L}'$, respectively. It is shown that distillation with $\mathcal{L}_{MSE}$, that is, direct logit distillation without hindering term δ_∞, is the best training scheme for various teacher-student pairs. We also found the consistent improvements in ensemble distillation [Hinton *et al.*, 2015]. For the ensemble distillation using MSE loss, an ensemble of logit predictions (i.e., an average of logit predictions) are used by multiple teachers. We obtained the test accuracy of WRN16-2 (75.60%) when the WRN16-4, WRN-28-4, and WRN-40-6 models were used as ensemble teachers in this manner. Moreover, the model trained with $\mathcal{L}_{MSE}$ has similar or better performance when compared to other existing KD methods, as described in Table 3.[1]

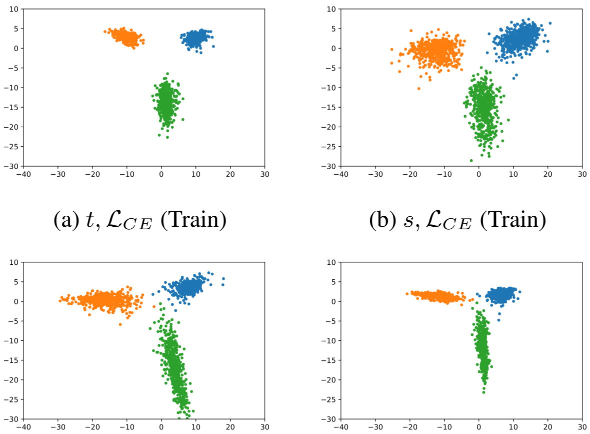
Furthermore, to measure the distance between the student's logit $\boldsymbol{z}^s$ and the teacher's logit $\boldsymbol{z}^t$ sample by sample, we describe the probabilistic density function (pdf) from the histogram for $\|\boldsymbol{z}^s - \boldsymbol{z}^t\|_2$ on the CIFAR-100 training dataset (Figure 4(a)). The logit distribution of the student with a large τ is closer to that of the teacher than with a small τ when $\mathcal{L}_{KL}$ is used. Moreover, $\mathcal{L}_{MSE}$ is more efficient in transferring the teacher's information to a student than $\mathcal{L}_{KL}$. Optimizing $\mathcal{L}_{MSE}$ aligns the student's logit with the teacher's logit. On the other hand, when τ becomes significantly large, $\mathcal{L}_{KL}$ has the δ_∞, and optimizing δ_∞ makes the student's logit mean deviate from that of the teacher's logit mean.

We further investigate the effect of δ_∞ on the penultimate layer representations (i.e., pre-logits). Based on $\delta_\infty \approx -\frac{1}{2K^2}(\sum_{j=1}^{K} \boldsymbol{z}_j^s)^2$, we can reformulate Eq. (8). Let $\boldsymbol{r}^s \in \mathbb{R}^d$ be the penultimate representation of student s from an instance $\boldsymbol{x}$, and $W^s \in \mathbb{R}^{K \times d}$ be the weight matrix of the student's fully connected layer. Then,

$$
\begin{aligned}
\delta_\infty &\approx -\frac{1}{2K^2}\left(\sum_{j=1}^{K} \boldsymbol{z}_j^s\right)^2 = -\frac{1}{2K^2}\left(\sum_{j=1}^{K}\sum_{n=1}^{d} W_{j,n}^s \boldsymbol{r}_n^s\right)^2 \\
&= -\frac{1}{2K^2}\left(\sum_{n=1}^{d} \boldsymbol{r}_n^s \sum_{j=1}^{K} W_{j,n}^s\right)^2 \\
&\geq -\frac{1}{2K^2}\left(\sum_{n=1}^{d}(\sum_{j=1}^{K} W_{j,n}^s)^2\right)\left(\sum_{n=1}^{d} {\boldsymbol{r}_n^s}^2\right) \\
&\quad (\because \text{Cauchy-Schwartz inequality}) \\
&= -\frac{1}{2K^2}\|\boldsymbol{r}^s\|_2^2 \left(\sum_{n=1}^{d}(\sum_{j=1}^{K} W_{j,n}^s)^2\right)
\end{aligned}
\tag{10}
$$

As derived in Eq. (10), training a network with $\mathcal{L}_{KL}$ encourages the pre-logits to be dilated via δ_∞ (Figure 4b). For visualization, following [Müller *et al.*, 2019], we first find an orthonormal basis constructed from the templates (i.e., the mean of the representations of the samples within the same class) of the three selected classes (apple, aquarium fish, and baby in our experiments). Then, the penultimate layer representations are projected onto the hyperplane based on the identified orthonormal basis. WRN-28-4 (t) is used as a teacher, and WRN-16-2 (s) is used as a student on the CIFAR-100 training dataset. As shown in the first row of Figure 5, when WRN-like architectures are trained with $\mathcal{L}_{CE}$ based on ground-truth hard targets, clusters are tightened as the model's complexity increases. As shown in the second row of Figure 5, when the student s is trained with $\mathcal{L}_{KL}$ with infinite τ or with $\mathcal{L}_{MSE}$, both representations attempt to follow the shape of the teacher's representations but differ in the degree of cohesion. This is because δ_∞ makes the pre-logits become much more widely clustered. Therefore,

[1] We excluded the additional experiments for the replacement with MSE loss in feature-based distillation methods. It is difficult to add the MSE loss or replace the KL loss with MSE loss in the existing works because of the sensitivity to hyperparameter optimization. Their methods included various types of hyperparameters that need to be optimized for their settings.



(a) $t, \mathcal{L}_{CE}$ (Train)

(b) $s, \mathcal{L}_{CE}$ (Train)

(c) $s, \mathcal{L}_{KL}(\tau = \infty)$ (Train)

(d) $s, \mathcal{L}_{MSE}$ (Train)

Figure 5: Visualizations of pre-logits on CIFAR-100 according to the change of loss function. Here, we use the classes "apple," "aquarium fish," and "baby." t indicates the teacher network (WRN-28-4), and s indicates the student network (WRN-16-2).

Student	$\mathcal{L}_{KL}$						$\mathcal{L}_{MSE}$
	τ=0.1	τ=0.5	τ=1	τ=5	τ=20	τ=∞	
WRN-16-2	51.64	52.07	51.36	50.11	49.69	49.46	49.20

Table 4: Top-1 test accuracies on CIFAR-100. WRN-28-4 is used as a teacher for $\mathcal{L}_{KL}$ and $\mathcal{L}_{MSE}$. Here, the teacher (WRN-28-4) was not fully trained. The training accuracy of the teacher network is 53.77%.

Student	$\mathcal{L}_{CE}$	$\mathcal{L}_{KL}$ (Standard)	$\mathcal{L}_{KL}$ (τ=20)	$\mathcal{L}_{MSE}$
ResNet-50	76.28	77.15	77.52	75.84

Table 5: Test accuracy on the ImageNet dataset. We used a (teacher, student) pair of (ResNet-152, ResNet-50). We include the results of the baseline and $\mathcal{L}_{KL}$ (standard) from [Heo *et al.*, 2019a]. The training accuracy of the teacher network is 81.16%.

$\mathcal{L}_{MSE}$ can shrink the representations more than $\mathcal{L}_{KL}$ along with the teacher.

4.4 Effects of a Noisy Teacher

We investigate the effects of a noisy teacher (i.e., a model poorly fitted to the training dataset) according to the objective. It is believed that the *label matching* ($\mathcal{L}_{KL}$ with a small τ) is more appropriate than the *logit matching* ($\mathcal{L}_{KL}$ with a large τ or the $\mathcal{L}_{MSE}$) under a noisy teacher. This is because *label matching* neglects the negative information of the outputs of an untrained teacher. Table 4 describes top-1 test accuracies on CIFAR-100, where the used teacher network (WRN-28-4) has a training accuracy of 53.77%, which is achieved in 10 epochs. When poor knowledge is distilled, the students following the *label matching* scheme performed better than the students following the *logit matching* scheme, and the extreme *logit matching* through $\mathcal{L}_{MSE}$ has the worst performance. Similarly, it seems that *logit matching* is not suitable for large-scale tasks. Table 5 presents top-1 test accuracies on ImageNet, where the used teacher network (ResNet-152) has a training accuracy of 81.16%, which is provided in PyTorch. Even in this case, the extreme *logit matching* exhibits the worst performance. The utility of negative logits (i.e., negligible aspect when τ is small) was discussed in [Hinton *et al.*, 2015].

5 Sequential Distillation

In [Cho and Hariharan, 2019], the authors showed that more extensive teachers do not mean better teachers, insisting that the capacity gap between the teacher and the student is a more important factor than the teacher itself. In their results, using a medium-sized network instead of a large-scale network as a teacher can improve the performance of a small network by reducing the capacity gap between the teacher and the student. They also showed that sequential KD (large network → medium network → small network) is not conducive to generalization when $(\alpha, \tau) = (0.1, 4)$ in Eq. (2). In other words, the best approach is a direct distillation from the medium model to the small model.

Table 6 describes the test accuracies of sequential KD, where the largest model is WRN-28-4, the intermediate model is WRN-16-4, and the smallest model is WRN-16-2. Similar to the previous study, when $\mathcal{L}_{KL}$ with $\tau = 3$ is used to train the small network iteratively, the direct distillation from the intermediate network to the small network is better (i.e., WRN-16-4 → WRN-16-2, 74.84%) than the sequential distillation (i.e., WRN-28-4 → WRN-16-4 → WRN-16-2, 74.52%) and direct distillation from a large network to a small network (i.e., WRN-28-4 → WRN-16-2, 74.24%). The same trend occurs in $\mathcal{L}_{MSE}$ iterations.

On the other hand, we find that the medium-sized teacher can improve the performance of a smaller-scale student when $\mathcal{L}_{KL}$ and $\mathcal{L}_{MSE}$ are used sequentially (the last fourth row) despite the large capacity gap between the teacher and the student. KD iterations with such a strategy might compress the model size more effectively, and hence should also be considered in future work. Furthermore, our work is the first study on the sequential distillation at the *objective* level, not at the *architecture* level such as [Cho and Hariharan, 2019; Mirzadeh *et al.*, 2020].

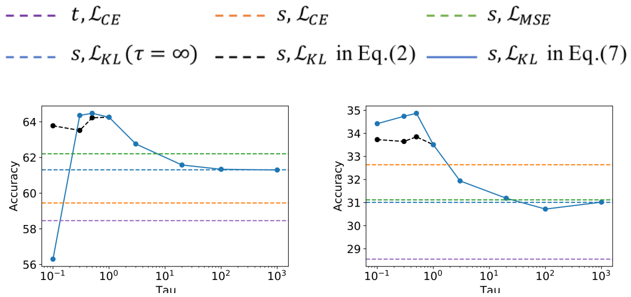
6 Robustness to Noisy Labels

In this section, we investigate how *noisy labels*, samples annotated with incorrect labels in the training dataset, affect the distillation ability when training a teacher network. This setting is related to the capacity for memorization and generalization. Modern deep neural networks even attempt to memorize samples perfectly [Zhang *et al.*, 2016]; hence, the teacher might transfer *corrupted knowledge* to the student in this situation. Therefore, it is thought that logit matching might not be the best strategy when the teacher is trained using a noisy label dataset.

From this insight, we simulate the noisy label setting to evaluate the robustness on CIFAR-100 by randomly flipping a certain fraction of the labels in the training dataset following

WRN-28-4	WRN-16-4	WRN-16-2	Test accuracy
X	X	$\mathcal{L}_{CE}$	72.68 %
X	$\mathcal{L}_{CE}$ (77.28%)	$\mathcal{L}_{KL}(\tau = 3)$	74.84 %
		$\mathcal{L}_{KL}(\tau = 20)$	75.42 %
		$\mathcal{L}_{MSE}$	75.58 %
$\mathcal{L}_{CE}$ (78.88%)	X	$\mathcal{L}_{KL}(\tau = 3)$	74.24 %
		$\mathcal{L}_{KL}(\tau = 20)$	75.15 %
		$\mathcal{L}_{MSE}$	75.54 %
$\mathcal{L}_{CE}$ (78.88%)	$\mathcal{L}_{KL}(\tau = 3)$ (78.76%)	$\mathcal{L}_{KL}(\tau = 3)$	74.52 %
		$\mathcal{L}_{KL}(\tau = 20)$	75.47 %
		$\mathcal{L}_{MSE}$	**75.78 %**
	$\mathcal{L}_{MSE}$ (79.03%)	$\mathcal{L}_{KL}(\tau = 3)$	74.83 %
		$\mathcal{L}_{KL}(\tau = 20)$	75.47 %
		$\mathcal{L}_{MSE}$	75.48 %

Table 6: Test accuracies of sequential knowledge distillation. In each entry, we note the objective function that used for the training. 'X' indicates that distillation was not used in training.



(a) Symmetric noise 40%

(b) Symmetric noise 80%

Figure 6: Test accuracy graph as τ changes on CIFAR-100. We use the (teacher, student) as (WRN-28-4, WRN-16-2).

a symmetric uniform distribution. Figure 6 shows the test accuracy graphs as the loss function changes. First, we observe that a small network (WRN-16-2 (s), orange dotted line) has a better generalization performance than an extensive network (WRN-28-4 (t), purple dotted line) when models are trained with $\mathcal{L}_{CE}$. This implies that a complex model can memorize the training dataset better than a simple model, but cannot generalize to the test dataset. Next, WRN-28-4 (purple dotted line) is used as the teacher model. When the noise is less than 50%, extreme logit matching ($\mathcal{L}_{MSE}$, green dotted line) and logit matching with δ_∞ ($\mathcal{L}_{KL}(\tau = \infty)$, blue dotted line) can mitigate the label noise problem compared with the model trained with $\mathcal{L}_{CE}$. However, when the noise is more than 50%, these training cannot mitigate this problem because it follows corrupted knowledge more often than correct knowledge.

Interestingly, the best generalization performance is achieved when we use $\mathcal{L}_{KL}$ with $\tau \leq 1.0$. In Figure 6, the blue solid line represents the test accuracy using the rescaled loss function from the black dotted line when $\tau \leq 1.0$. As expected, logit matching might transfer the teacher's overconfidence, even for incorrect predictions. However, the proper objective derived from both logit matching and label matching enables similar effects of label smoothing, as studied in [Lukasik *et al.*, 2020; Yuan *et al.*, 2020]. Therefore, $\mathcal{L}_{KL}$ with $\tau = 0.5$ appears to significantly mitigate the problem of noisy labels.

7 Conclusion

In this paper, we first showed the characteristics of a student trained with $\mathcal{L}_{KL}$ according to the temperature-scaling hyperparameter τ. As τ goes to 0, the trained student has the *label matching* property. In contrast, as τ goes to ∞, the trained student has the *logit matching* property. Nevertheless, $\mathcal{L}_{KL}$ with a sufficiently large τ cannot achieve complete logit matching owing to δ_∞. To achieve this goal, we proposed a direct logit learning framework using $\mathcal{L}_{MSE}$ and improved the performance based on this loss function. In addition, we showed that the model trained with $\mathcal{L}_{MSE}$ followed the teacher's penultimate layer representations more than that with $\mathcal{L}_{KL}$. We observed that sequential distillation can be a better strategy when the capacity gap between the teacher and the student is large. Furthermore, we empirically observed that, in the noisy label setting, using $\mathcal{L}_{KL}$ with τ near 1 mitigates the performance degradation rather than extreme logit matching, such as $\mathcal{L}_{KL}$ with $\tau = \infty$ or $\mathcal{L}_{MSE}$.

Acknowledgments

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [No.2019-0-00075, Artificial Intelligence Graduate School Program (KAIST)] and [No. 2021-0-00907, Development of Adaptive and Lightweight Edge-Collaborative Analysis Technology for Enabling Proactively Immediate Response and Rapid Learning].

References

[Aguinaldo *et al.*, 2019] Angeline Aguinaldo, Ping-Yeh Chiang, Alexander Gain, Ameya Patil, Kolten Pearson, and Soheil Feizi. Compressing gans using knowledge distillation. *CoRR*, abs/1902.00159, 2019.

[Ba and Caruana, 2013] Lei Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *arXiv preprint arXiv:1312.6184*, 2013.

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

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

[He *et al.*, 2016] 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*, pages 770–778, 2016.

[Heo *et al.*, 2019a] 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*, pages 1921–1930, 2019.

[Heo *et al.*, 2019b] 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, pages 3779–3787, 2019.

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

[Jang *et al.*, 2016] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

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

[Lukasik *et al.*, 2020] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 6448–6458, Virtual, 13–18 Jul 2020. PMLR.

[Maddison *et al.*, 2016] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *CoRR*, abs/1611.00712, 2016.

[Mirzadeh *et al.*, 2020] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5191–5198, 2020.

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

[Müller *et al.*, 2019] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4694–4703, 2019.

[Park *et al.*, 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*, pages 3967–3976, 2019.

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

[Romero *et al.*, 2014] 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.

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

[Szegedy *et al.*, 2016] 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*, pages 2818–2826, 2016.

[Tang *et al.*, 2020] 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.

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

[Urban *et al.*, 2016] Gregor Urban, Krzysztof J Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Rich Caruana, Abdelrahman Mohamed, Matthai Philipose, and Matt Richardson. Do deep convolutional nets really need to be deep and convolutional? *arXiv preprint arXiv:1603.05691*, 2016.

[Yuan *et al.*, 2020] 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.

[Zagoruyko and Komodakis, 2016a] 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*, 2016.

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

[Zhang *et al.*, 2016] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

[Zhang *et al.*, 2019] 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.