Extractive Knowledge Distillation

Takumi Kobayashi
National Institute of Advanced Industrial Science and Technology
1-1-1 Umezono, Tsukuba, Japan
takumi.kobayashi@aist.go.jp

Abstract

Knowledge distillation (KD) transfers knowledge of a teacher model to improve performance of a student model which is usually equipped with lower capacity. In the KD framework, however, it is unclear what kind of knowledge is effective and how it is transferred. This paper analyzes a KD process to explore the key factors. In a KD formulation, softmax temperature entangles three main components of student and teacher probabilities and a weight for KD, making it hard to analyze contributions of those factors separately. We disentangle those components so as to further analyze especially the temperature and improve the components respectively. Based on the analysis about temperature and uniformity of the teacher probability, we propose a method, called extractive distillation*, for extracting effective knowledge from the teacher model. The extractive KD touches only teacher knowledge, thus being applicable to various KD methods. In the experiments on image classification tasks using Cifar-100 and TinyImageNet datasets, we demonstrate that the proposed method outperforms the other KD methods and analyze feature representation to show its effectiveness in the framework of transfer learning.*

1. Introduction

Neural networks are widely applied to various fields of pattern classification [9, 26]. The state-of-the-art networks are equipped with huge amount of parameters to produce remarkable performance, even surpassing human. It, however, is hard to deploy the high-capacity networks to such as edge devices of limited computing resources.

Knowledge distillation (KD) [2, 12] is a distinctive approach to enhance the performance of low-capacity networks with a help of pre-trained high-capacity models. KD transfers knowledge from a cumbersome model (teacher) to a single small model (student) [12]; in other words, the high-capacity network teaches the low-capacity network how to optimize parameters like a *teacher-student* model. KD provides a general framework such that we can train a low-capacity network from scratch while improving performance. It also complements the other approaches [19].

A vanilla KD [12] is simply formulated through matching two (softened) softmax probabilities of the student and teacher neural networks, surprisingly working well in comparison even to its sophisticated variants [29]. Despite its simplicity, it has not been completely analyzed/understood what kind of information is distilled from a teacher model and how it is propagated toward a student to improve performance. There are some works to analyze the KD process [6, 28, 34, 3, 14] for revealing the intrinsic characteristics of KD based on the vanilla KD formulation [12]. As shown in Fig. 1, however, the simple formulation is composed of KD components *entangled* via a softmax *temperature*, making it hard to analyze those components separately. In this work, we disentangle those KD factors by considering a general formulation of KD to proceed further analysis while giving a light on a new perspective of the KD process. Based on the analysis, we propose a novel method, called *extractive knowledge distillation*, to extract further effective teacher knowledge for improving performance in a simple framework. It is thus applicable to various KD methods that are based on matching softmax probabilities.

Our contributions are summarized as follows.

- We introduce a general formulation for KD to disentangle three KD components which are connected via temperature in a vanilla KD formulation [12]. Those disentangled three components are separately analyzed, especially in terms of temperature.
- The teacher probability is viewed from a novel perspective regarding temperature and uniformity to propose an *extractive distillation* process for further extracting effective knowledge, while other methods directly employ *raw* teacher probabilities as knowledge.
- The proposed extractive KD is empirically evaluated by using CNNs on image classification tasks of Cifar-100 and TinyImageNet through comparison to the other methods. The method is also analyzed from a feature representation viewpoint in transfer learning.

2. Related Works

Since knowledge distillation (KD) was proposed in the seminal works [2, 12], it attracts keen attention in various fields related to machine learning. While a vanilla KD [12] is formulated to match logits of student and teacher models, there are variants to extend KD such as by matching intermediate layer representations [24, 10, 11], feature map attention [37], ensemble models [17] and relationships among samples [21]. The other techniques, e.g., sophisticated self-training [32] and data augmentation [35], are also incorporated into the KD framework to further boost performance [29, 33, 31]. KD is applied to compare not only teacher-student models of different capacities but also same capacity models in a self-distillation framework [38, 6]. Comprehensive survey for KD variants can be found in [8].

While the KD has been extended in various ways with promising performance, it is still unclear what kind of knowledge is essentially transferred from a teacher to a student. KD is theoretically analyzed regarding such as transfer risk bound through linearization [23] or by means of neural tangent kernel [13]. Recently, there are some works to analyze the KD process in detail and explore its key factors [6, 28, 34, 3, 14] through empirical studies as follow.

In the KD framework, teacher networks are analyzed in terms of capacity and (training) optimality in [34] showing that lower-capacity and poorly-trained teacher networks even contribute to performance improvement. Discrepancy between teacher and student capacities is also analyzed in [3], exploring what kind of teacher model is beneficial for KD such as through early stopping. In [6], softmax probabilities of a teacher model are regarded as weighting training samples in a similar way to the importance weighting scheme. From the weighting viewpoint, the teacher probabilities are further analyzed in [28] to clarify gradient weighing in back-propagation. The softmax probabilities of a teacher and a student model are *softened* by a temperature τ mainly for effective matching [12]. From the viewpoint of logit matching, [14] analyzes how the temperature works on training a student model. In this work, we also focus on the temperature τ to clarify effective knowledge from the teacher logits towards performance improvement in a different approach from [14].

KD is also discussed through a connection to label smoothing regularization (LSR) [27, 20] since the KD loss is formulated in a similar form to LSR; KD is referred to as *adaptive* LSR in [28] or *learned* LSR in [34]. In contrast to those works, through disentangling KD factors, we find out the effect of label smoothing *within* the teacher probabilities and leverage it to improve the teacher knowledge. The refined teacher knowledge is effectively transferred to training a student model with performance improvement. Thus, our method contrasts with the other KDs which directly employ softened teacher probabilities to be transferred.

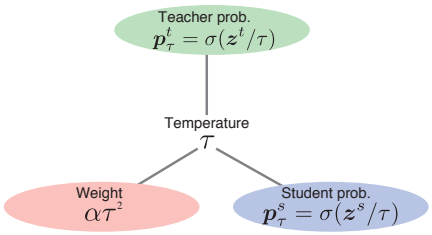


Figure 1. Entanglement via temperature τ.

3. Method

3.1. Disentangled KD formulation

Suppose we have a student model (neural network) ϕ^s and a teacher model ϕ^t pre-trained on a target dataset. Knowledge distillation (KD) is applied to train the student model through transferring knowledge of the teacher. For that purpose, a vanilla KD is formulated by matching the outputs of those two models in the following loss [12];

$$\mathcal{L} = (1-\alpha)\mathcal{H}(\mathrm{y}, \mathrm{p}^s) + \alpha\tau^2\mathcal{H}\big(\mathrm{p}_\tau^t, \mathrm{p}_\tau^s\big), \tag{1}$$

$$\frac{\partial \mathcal{L}}{\partial \mathrm{z}^s} = (1-\alpha)(\mathrm{p}^s - \mathrm{y}) + \alpha\tau(\mathrm{p}_\tau^s - \mathrm{p}_\tau^t), \tag{2}$$

where $\mathcal{H}(\mathrm{q}, \mathrm{p}) = -\sum_{c=1}^{C} \mathrm{q}[c] \log \mathrm{p}[c]$ measures cross-entropy between two probability distributions and $\mathrm{p}_\tau^s = \sigma(\mathrm{z}^s/\tau)$ and $\mathrm{p}_\tau^t = \sigma(\mathrm{z}^t/\tau)$ are softened softmax probabilities with a temperature τ based on the logits z^s and z^t produced by the student ϕ^s and teacher ϕ^t, respectively;

$$\mathrm{p}_\tau[c] = \sigma(\mathrm{z}/\tau)[c] = \frac{\exp(z_c/\tau)}{\sum_{i=1}^{C}\exp(z_i/\tau)}. \tag{3}$$

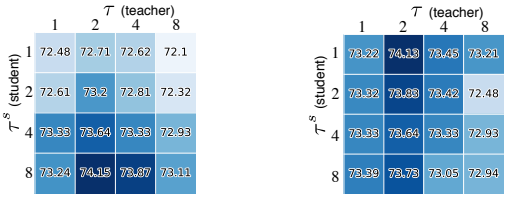
In (1), a classification loss (the first term) is simply measured by the cross entropy between $\mathrm{p}^s = \mathrm{p}_{\tau=1}^s = \sigma(\mathrm{z}^s)$ and the ground-truth (one-hot) label y. KD is essentially embedded in the second regularization term, being isolated from the classification term; those are mixed via a parameter α.

In the KD term, the temperature τ is involved in three components of the student softmax probability p_τ^s, the teacher probability p_τ^t and the weight of the KD term $\alpha\tau^2$, as shown in Fig. 1. The entanglement makes it difficult to analyze those factors individually and in particular to explore a key role of the temperature in KD. Therefore, we *disentangle* the three components by reformulating a vanilla KD (1) into the following general one;

$$\mathcal{L}_g = \gamma\mathcal{H}(\mathrm{y}, \mathrm{p}^s) + \beta\tau^s\mathcal{H}\big(\mathrm{q}_\tau^t, \mathrm{p}_{\tau^s}^s\big), \tag{4}$$

$$\frac{\partial \mathcal{L}_g}{\partial \mathrm{z}^s} = \gamma(\mathrm{p}^s - \mathrm{y}) + \beta(\mathrm{p}_{\tau^s}^s - \mathrm{q}_\tau^t), \tag{5}$$

where we define a student probability $\mathrm{p}_{\tau^s}^s = \sigma(\mathrm{z}^s/\tau^s)$ with a student temperature τ^s separately from a teacher *annotation* q_τ^t which is defined by an arbitrary probabilistic form based on the teacher logit z^t with a teacher temperature τ. (4) is reduced to (1) by $\tau^s = \tau$, $\gamma = 1 - \alpha$,



(a) Standard weight ($\beta = 0.9\tau^s$) (b) Disentangled weight ($\beta = 3.6$)

Figure 2. Classification accuracies (%) on Cifar-100 [16] by various student temperature τ^s and teacher temperature τ^t under the two weighting scenarios of standard (1) and disentangled (4) ones.

$\beta = \alpha\tau$ and $\mathrm{q}^t_\tau = \sigma(\mathbf{z}^t/\tau)$; note that the standard setting of $\{\alpha = 0.9, \tau = 4\}$ in (1) produces $\{\gamma = 0.1, \beta = 3.6\}$ in (4). Based on the general formulation, we analyze a KD process from the following three aspects in Sec. 3.2∼3.4.

3.2. Student probability

We here set the teacher annotation as $\mathrm{q}^t_\tau = \sigma(\mathbf{z}^t/\tau)$ which is further analyzed in the next section. The disentanglement in (4) allows us to *respectively* apply two types of temperatures τ^s and τ to the student $\mathrm{p}^s_{\tau^s} = \sigma(\mathbf{z}^s/\tau^s)$ and the teacher $\mathrm{q}^t_\tau = \sigma(\mathbf{z}^t/\tau)$, while an *identical* temperature τ is shared in a vanilla KD formulation (1).

We conduct preliminary experiments to evaluate the disentanglement regarding the student and the teacher probabilities in terms of temperatures. KD is applied to image classification on Cifar-100 dataset [16] using ResNet32x4 [9] (teacher) and ResNet8x4 (student); the detailed experimental setting is shown in Sec. 4. Classification accuracies are measured over various temperatures under two weighting scenarios of $\beta = 0.9\tau^s$ and $\beta = 3.6$; the former one is the ordinary setting in (1) with $\alpha = 0.9$ while the latter excludes the temperature from the weight in (4) assuming a standard setting of $\tau^s = 4$.

As shown in Fig. 2a, the classification performance is affected by *both* the student and teacher temperatures in the ordinary weighting. On the other hand, the disentangled weighting ($\beta = 0.9$) in Fig. 2b clarifies respective contributions of those two temperatures. We find that performance is dominated by the *teacher* temperature τ while the student one τ^s is less influential in the performance. This can also be viewed in the following theoretical way.

The tempered softmax loss (4) is rewritten to

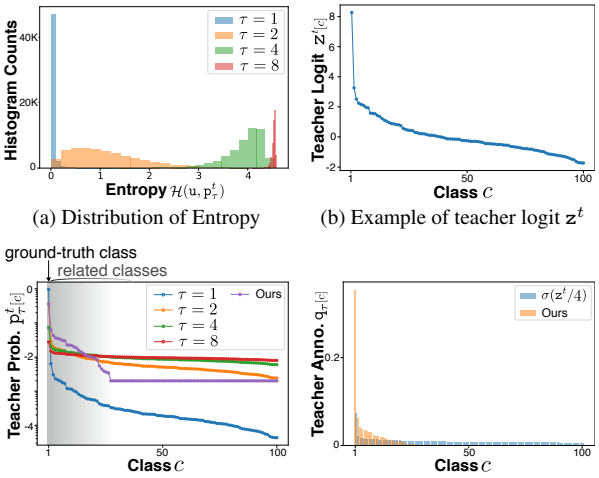
$$\mathcal{L}_g = \gamma\mathcal{H}(\mathrm{y}, \sigma(\tau^s\tilde{\mathbf{z}}^s)) + \beta\tau^s\mathcal{H}\big(\mathrm{q}^t_\tau, \sigma(\tilde{\mathbf{z}}^s)\big), \qquad (6)$$

where $\tilde{\mathbf{z}}^s = \mathbf{z}^s/\tau^s$. Under the fixed teacher annotation q^t_τ, the student temperature τ^s operates on the classifier loss, affecting training dynamics as analyzed in the tempered softmax loss [1] rather than significant performance improvement. Thus, the student temperature τ^s is less contributive to the performance in comparison to the teacher one τ which directly controls the teacher annotation (Sec. 3.3).

Based on the analysis, without loss of performance, we employ $\tau^s = 1$ to simplify the KD formulation (4) into

$$\mathcal{L}_g = \gamma\mathcal{H}(\mathrm{y}, \mathrm{p}^s) + \beta\mathcal{H}\big(\mathrm{q}^t_\tau, \mathrm{p}^s\big), \qquad (7)$$

where a temperature τ appears only on the teacher q^t_τ.



(a) Distribution of Entropy (b) Example of teacher logit $\mathbf{z}^t$

(c) Teacher probability p^t_τ [$\log_{10}$] (d) Teacher annotation [linear-scale]

Figure 3. Characteristics of teacher probability $\mathrm{p}^t_\tau = \sigma(\mathbf{z}^t/\tau)$ on various temperature τ. Our annotation (9) in (c,d) is implemented by $\tau = 4, \epsilon = 0.2$.

3.3. Teacher annotation

As shown above, the central role of KD is to extract *knowledge* from the teacher logit $\mathbf{z}^t$ through a temperature τ and to transfer it in the form of q^t_τ in (7). To that end, one usually resorts to a higher temperature $\tau > 1$ for distilling such knowledge by means of the softened softmax probabilities $\sigma(\mathbf{z}^t/\tau)$. We analyze effect of the teacher temperature τ for constructing an effective teacher annotation q^t_τ.

3.3.1 Effect of higher temperature

Fig. 3a shows distribution of entropy computed on a teacher probability $\mathrm{p}^t_\tau = \sigma(\mathbf{z}^t/\tau)$ by $\mathcal{H}(\mathrm{u}, \mathrm{p}^t_\tau)$ where $\mathrm{u} = \{\mathrm{u}[c] = \frac{1}{C}, \forall c\}$. The lower temperature, e.g., $\tau = 1$, produces too sparse signals while the higher one, e.g., $\tau = 8$, smooths out the probability toward uniform. A softened softmax probability *blurs* knowledge (Fig. 3cd) derived from a teacher logit $\mathbf{z}^t$ which distinctively activates the ground-truth class (Fig. 3b). The blurring favorably reveals the following two characteristics which are effective to train a student model.

Relationship among class categories. While the ground-truth class usually receives a prominent probability at any temperatures, as shown in Fig. 3c, the moderately high temperature activates the probabilities of the non ground-truth classes which are suppressed in the low-tempered softmax probability. Those activations accompanying the ground-truth class reflect the similarities among the class categories [28], i.e, class relationships, which can be regarded as the core information of dark knowledge [12]. The class

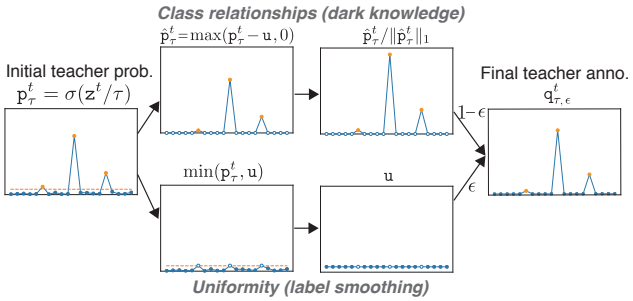


Figure 4. Extractive distillation of teacher knowledge. It first decompose the teacher probability $\sigma(\mathbf{z}^t/\tau)$ into two parts of class relationships and uniformity. Then, the decomposed factors are favorably combined with a parameter ϵ.

relationships encoded by the teacher probabilities encourage the student model to capture the intrinsic characteristics of recognition targets to improve feature representation.

Uniformity. On the other hand, the higher temperature activates probabilities of *all* the classes including not only the related classes but also the other irrelevant ones (Fig. 3cd) since the teacher probability approaches to uniform $\mathrm{p}_\tau^t = \sigma(\mathbf{z}^t/\tau) \to \mathrm{u}$ as $\tau \to \infty$ (Fig. 3a). The uniformity across classes is heuristically employed as the label smoothing regularization (LSR) [27, 20]. From this viewpoint, the higher temperature enhances the regularization effect of KD. The LSR is effectively applied to such as classification tasks [27] by regularizing feature representation via large-margin effects [20, 15]. In the KD literature, the connection to label smoothing has been discussed from the viewpoint of formulation [28, 34], and the LSR effect *embedded* in the teacher probability p_τ^t has not been clearly discussed.

3.3.2 Extractive distillation of teacher knowledge

The success of KD would be built on the above-mentioned two characteristics to enhance feature representation as well as to regularize the training toward high generalization performance. They, however, are again *entangled* via the teacher temperature τ. Namely, the single parameter τ simultaneously controls both effects of class relationships and uniformity, thereby being unable to properly balance them; for example, it is difficult to extract broader class similarities while suppressing uniformity (low label smoothing regularization) since the higher temperature increases both the range of class relationships and the uniformity as shown in Fig. 3c. Thus, for improving the teacher annotation beyond a simple softened probability $\sigma(\mathbf{z}^t/\tau)$, we disentangle those two characteristics in the teacher probability as follows (Fig. 4); we call this process *extractive distillation*[1].

First, for extracting class relationships, the teacher probability $\mathrm{p}_\tau^t = \sigma(\mathbf{z}^t/\tau)$ is decomposed on the basis of the *uniform probability* $\mathrm{u} = \{\mathrm{u}[c] = \frac{1}{C}\}_{c=1}^C$ by

$$\mathrm{p}_\tau^t = \max(\mathrm{p}_\tau^t - \mathrm{u}, 0) + \min(\mathrm{p}_\tau^t, \mathrm{u}) = \hat{\mathrm{p}}_\tau^t + \min(\mathrm{p}_\tau^t, \mathrm{u}), \quad (8)$$

where the max and min operators are applied in an element-wise manner. In the decomposition (8), the distinctive probabilities $\hat{\mathrm{p}}_\tau^t = \max(\mathrm{p}_\tau^t - \mathrm{u}, 0)$ are discriminated from the other subtle ones through thresholding by u. $\hat{\mathrm{p}}_\tau^t$ is defined as the deviation from the uniform distribution u and thereby sparsely encodes class relationships even by higher τ. On the other hand, the other remaining probabilities $\min(\mathrm{p}_\tau^t, \mathrm{u})$ are regarded as rather *noisy* ones. Those noises contribute to label smoothing regularization as described above and can be approximated by the uniform probability u. Thereby, the class relationships and uniformity are separated as shown in Fig. 4. The decomposed two factors are finally merged into the teacher annotation $\mathrm{q}_{\tau,\epsilon}^t$ by

$$\mathrm{q}_{\tau,\epsilon}^t = (1-\epsilon)\frac{\hat{\mathrm{p}}_\tau^t}{\|\hat{\mathrm{p}}_\tau^t\|_1} + \epsilon\,\mathrm{u}, \quad (9)$$

where ϵ is a parameter as in the label smoothing [27].

In this extractive distillation process (Fig. 4), we separate the effects of class relationships and uniformity (label smoothing), both of which contribute to performance improvement, yet being heavily entangled via the temperature τ in the teacher probabilities used in ordinary KD. Then, those are favorably mixed up in the form (9) to introduce the LSR effect in a *controllable* way by means of the parameter $\epsilon \in [0, 1]$. We can intuitively construct the teacher annotation $\mathrm{q}_{\tau,\epsilon}^t$ based on the two parameters τ and ϵ; it is possible to encode broad class relationships by higher τ while introducing a bit LSR effect via small ϵ, as shown in Fig. 3cd.

3.3.3 Discussion

We can reconstruct the teacher probability p_τ^t (8) by

$$\mathrm{p}_\tau^t \approx \hat{\mathrm{p}}_\tau^t + (1 - \|\hat{\mathrm{p}}_\tau^t\|_1)\mathrm{u}, \quad (10)$$

where we equally distribute the (rest) probability mass, $1 - \|\hat{\mathrm{p}}_\tau^t\|_1$, to all the classes. The approximated form (10) is involved in our framework (9) by *adaptively* setting $\epsilon = 1 - \|\hat{\mathrm{p}}_\tau^t\|_1$ across samples while we consider the *constant* ϵ in (9). From this viewpoint, KD methods which directly employ *raw* teacher probability p_τ^t inject label smoothing effect variably across samples according to the classification confidence $\|\hat{\mathrm{p}}_\tau^t\|_1$ of the teacher model.

In KD-top$k$ [28], the class relationships are extracted by top-k probabilities in a similar manner to (8) and the rest probability mass is shared across the other classes as in (10);

$$\mathrm{q}_{\tau,k}^t = \hat{\mathrm{p}}_{\tau,k}^t + (1 - \|\hat{\mathrm{p}}_{\tau,k}^t\|_1)\check{\mathrm{u}}_k, \quad (11)$$

$$\hat{\mathrm{p}}_{\tau,k}^t[c] = \begin{cases} \mathrm{p}_\tau^t[c] & c \in \mathcal{K} \\ 0 & c \notin \mathcal{K} \end{cases}, \quad \check{\mathrm{u}}_k[c] = \begin{cases} 0 & c \in \mathcal{K} \\ \frac{1}{C-k} & c \notin \mathcal{K} \end{cases}, \quad (12)$$

[1] We introduce u to disentangle the two properties and then distill effective knowledge as in the chemical process of *extractive distillation* [7].

where $\mathcal{K}$ indicates an index set of top-k probabilities, $|\mathcal{K}| = k$. Our approach (9) is clearly different from the KD-top$k$ (11) in the following two points: (1) we identify the class relationships ($\hat{\mathrm{p}}^t_\tau$) based on the deviation from uniformity u and (2) modify the uniformity, i.e., LSR effect, via a *fixed* parameter ϵ. These two points are detailed below.

(1) The top-k components are firmly picked up no matter how the logits z^t are distributed; it might miss distinctive activations by smaller k. Thus, the parameter k should be carefully tuned on tasks and datasets [28]. In contrast, our approach (8) adaptively picks up the distinctive components based on the logit distribution on the basis of uniformity;

$$\mathrm{p}^t_\tau[c] > \frac{1}{C} \Leftrightarrow \frac{\mathrm{z}^t[c]}{\tau} > \log\Big[\sum_{i=1}^{C} \exp\Big(\frac{\mathrm{z}^t[i]}{\tau}\Big)\Big] - \log(C),$$

where the log-sum-exp term reflects the statistical property of the logits z^t/τ, being related to the class relationships.

(2) The LSR effect is *constantly* introduced by a small constant fraction ϵ in (9). On the other hand, the adaptive ϵ in (11) dependent on the probability mass $\|\hat{\mathrm{p}}^t_{\tau,k}\|_1$ directly reflects the uncertainty of the teacher decision which is derived from the teacher model itself rather apart from the natural characteristics (intrinsic relationships) of the class categories. Thus, we remove the interference by the teacher model uncertainty and leverage constant uniformity to enhance generalization performance as in LSR [27, 20].

3.4. Weighting

Through the disentanglement (4), the weight β of the KD term can be flexibly set while a vanilla KD usually relates it to the temperature τ. Looking at the experimental results in Fig. 2b, the weight would also affect performance; the performance at $\tau^s = 1$ is improved by $\beta = 3.6$ (Fig. 2b) compared to $\beta = 0.9 \times 1 = 0.9$ (Fig. 2a), and at $\tau^s = 8$ the performance of $\beta = 0.9 \times 8 = 7.2$ (Fig. 2a) outperforms that of $\beta = 3.6$ (Fig. 2b).

To explore effective weighting, we focus on the back-propagation process. The gradients derived from the classification loss and the KD term are given by, respectively,

$$\frac{\partial \mathcal{L}_{cls}}{\partial \mathrm{z}^s} = \frac{\partial \mathcal{H}(\mathrm{y}, \mathrm{p}^s)}{\partial \mathrm{z}^s} = \mathrm{p}^s - \mathrm{y}, \; \frac{\partial \mathcal{L}_{kd}}{\partial \mathrm{z}^s} = \frac{\partial \mathcal{H}(\mathrm{q}^t_{\tau,\epsilon}, \mathrm{p}^s)}{\partial \mathrm{z}^s} = \mathrm{p}^s - \mathrm{q}^t_{\tau,\epsilon}, \tag{13}$$

the magnitudes (norms) of which are highly biased as shown in Fig. 8a; $\|\frac{\partial \mathcal{L}_{kd}}{\partial \mathrm{z}^s}\|_2 \ll \|\frac{\partial \mathcal{L}_{cls}}{\partial \mathrm{z}^s}\|_2$. Thus, the weight β should be *larger* to compensate the bias so that the two kinds of information are properly mixed. It can be determined through empirical evaluation (Fig. 8b).

In summary, we have disentangled three main KD factors of student and teacher probabilities and KD weight to improve them respectively in Sec. 3.2∼3.4; the parameter values are determined by ablation experiments in Sec. 4.1.

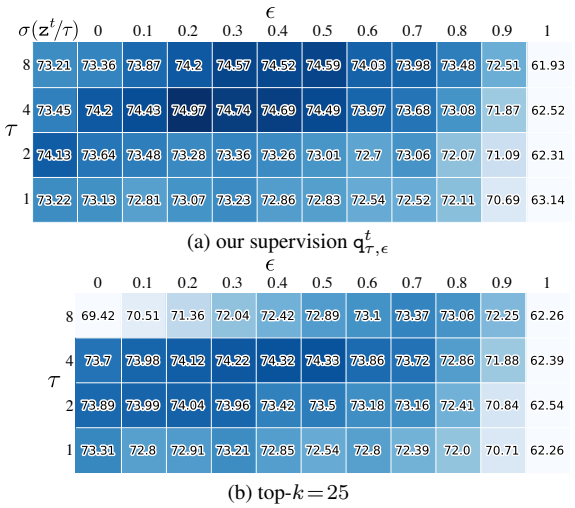


Figure 5. Performance results on Cifar-100 by using (a) our supervision $\mathrm{q}^t_{\tau,\epsilon}$ (9) on various temperature τ and ϵ. (b) It is compared to the top-$k = 25$ approach [28] which detects distinctive probabilities by top-k instead of $\hat{\mathrm{p}}^t_\tau$ (8).

4. Experimental Results

The proposed extractive KD method is applied to image classification tasks using CNNs. We follow the standard training protocol [29] of mini-batch 64, weight decay 0.0005 and SGD optimizer with momentum 0.9 and initial learning rate 0.05 which is divided by 10 at 150, 180 and 210-th epochs over total 240 epochs. Our KD is implemented in the general form (4) using $\tau^s = 1, \gamma = 0.1$ while the teacher annotation $\mathrm{q}^t_{\tau,\epsilon}$ (9) and the weight β are determined based on the ablation experiments in Sec. 4.1.

4.1. Ablation study

We first empirically analyze the teacher annotation $\mathrm{q}^t_{\tau,\epsilon}$ (9) in an ablation manner on Cifar-100 dataset [16] with $\beta = 3.6$. The Cifar-100 dataset provides a classification task of 32×32 images sampled from 100 object classes with 50,000 training samples as well as 10,000 test samples.

Parameters τ and ϵ . Fist, we analyze the temperature τ and the uniformity parameter ϵ in (9). Fig. 5a shows performance results on $\tau \in \{1, 2, 4, 8\}$ and $\epsilon \in [0, 1]$ in comparison to the teacher probability $\mathrm{q}^t_\tau = \sigma(\mathrm{z}^t/\tau)$. On the basis of $\epsilon = 0$, it is possible to identify the effects of the class relationships $\hat{\mathrm{p}}^t_\tau$ (8) and the uniformity u. The annotation $\mathrm{q}^t_{\tau,\epsilon}$ with $\epsilon = 0$ is solely composed of $\hat{\mathrm{p}}^t_\tau$ and the performance at $\epsilon = 0$ implies that the class relationship is effectively extracted by $\tau = 4$. On the other hand, the performance of $\sigma(\mathrm{z}^t/\tau)$ is degraded by $\tau > 2$ from the peak at $\tau = 2$. The discrepancy is due to the effect of uniformity related to label smoothing regularization (LSR). As shown in Fig. 3a, the uniformity of $\sigma(\mathrm{z}^t/\tau)$ is favorably small at $\tau = 2$ to enhance generalization performance but is too large by $\tau > 2$, imposing intense regularization on the student model to deteriorate performance though dark knowl-

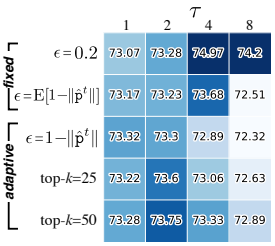


Figure 6. Comparison among various types of ϵ.

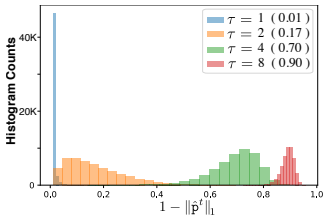
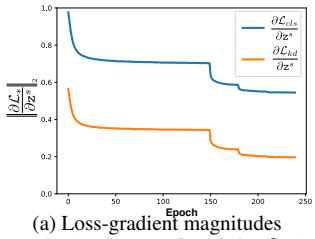
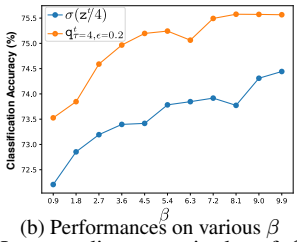


Figure 7. Distributions of the rest probability mass $1 - \|\hat{\mathbf{p}}_\tau^t\|_1$. The mean $E[1 - \|\hat{\mathbf{p}}_\tau^t\|_1]$ is shown in parentheses.



(a) Loss-gradient magnitudes



(b) Performances on various β

Figure 8. Analyses of weight β. (a) Loss-gradient magnitudes of $\mathcal{L}_{cls}$ and $\mathcal{L}_{kd}$ in (13) during 240 training epochs. (b) Classification accuracies on Cifar-100 over various β with two kinds of teacher annotations.

edge is favorably extracted by $\tau = 4$. Actually, by properly controlling the uniformity of $\mathbf{q}_{\tau,\epsilon}^t$ such as by $\epsilon = 0.2$, the performance is significantly improved; $73.45\% \rightarrow 74.97\%$ at $\tau = 4$. On the other hand, the larger ϵ also imposes too much regularization on training, thereby degrading performance. Thus, in the teacher annotation (9), *dark knowledge* regarding class relationships are effectively extracted via large temperature τ while injecting LSR effect by small ϵ to boost performance. Based on the experimental results, we apply $\{\tau = 4, \epsilon = 0.2\}$ to construct $\mathbf{q}_{\tau,\epsilon}^t$ in (9).

Comparison. Next, our extractive distillation model is compared to the competitive ones mentioned in Sec. 3.3.3. The similar model is given in (10) which *adaptively* assigns the rest probability mass $1 - \|\hat{\mathbf{p}}_\tau^t\|_1$ to ϵ at each sample for reconstructing $\mathbf{p}_\tau^t$. The distribution of $1-\|\hat{\mathbf{p}}_\tau^t\|_1$ across samples is shown in Fig. 7, indicating that $\tau = 1$ provides no LSR effect ($\epsilon \approx 0$) while $\tau = 8$ endows significant effect ($\epsilon \approx 1$); it is related to the uniformity shown in Fig. 3a. As shown in Fig. 6 (3rd row), the stronger regularization of the larger τ significantly deteriorates performance; it is noteworthy that the class relationship can be encoded even by the large $\tau = 8$ since it produces favorable performance by $\epsilon = 0.2$ (1st row in Fig. 6). For further comparison, we also apply the method that fixes ϵ as $E[1 - \|\hat{\mathbf{p}}_\tau^t\|_1]$, the *mean* across training samples, whose actual values are shown in parentheses in Fig. 7. The performance comparison in Fig. 6 between the adaptive and fixed ϵ clarifies that the adaptive ϵ over samples is not contributive to performance. As discussed in Sec. 3.3.3, $1-\|\hat{\mathbf{p}}_\tau^t\|_1$ reflects merely the prediction uncertainty derived from the teacher model and is irrelevant to encoding the intrinsic class relationships. The KD-top$k$ methods [28] work similarly to the adaptive model (10) since the top-k approach (11) also redistribute the rest probability mass to classes in the same way as (10). It is also possible to incorporate the top-k approach into our framework (Fig. 4) with a parameter ϵ by replacing $\hat{\mathbf{p}}_\tau^t = \max(\mathbf{p}_\tau^t, \mathbf{u})$ with $\hat{\mathbf{p}}_{\tau,k}^t$ (12) to detect class relationships. The performance results are shown in Fig. 5b, compared to ours in Fig. 5a. Our method outperforms the top-k approach even in the proposed framework (9), demonstrating that the thresholding (8) based on $\mathbf{u}$ is more effective than the sorting based on top-k as discussed in Sec. 3.3.3.

Weighting. THen, we evaluate the weight β according to the discussion in Sec. 3.4. Performance results across various β are shown in Fig. 8b. The performance is further improved by $\beta = 7.2$ in comparison to $\beta = 3.6$ which is based on the standard weighting of $\beta = \alpha\tau$ with $\alpha = 0.9$ and $\tau = 4$ in (1). We also evaluate the performance of a standard setting which employs $\gamma = 0.1, \mathbf{q}_\tau^t = \sigma(\mathbf{z}^t/\tau^t), \tau^t = \tau^s = 4$ in (4). Tuning β also improves performance of the standard method in our general formulation, though it is still inferior to our KD equipped with the improved teacher annotation.

We summarize our setting in the general KD form (4) by

$$\boxed{\tau^s = 1, \gamma = 0.1, \beta = 7.2 \text{ with annotation } \mathbf{q}_{\tau=4,\epsilon=0.2}^t \ (9)}.$$

4.2. Performance comparison

The proposed method is compared with the other KD methods which are formulated by reducing discrepancy between the student and teacher models. We follow the evaluation protocol provided in [29] by applying diverse CNN models of various capacities [9, 36, 26, 25, 39, 18].

The performance results on Cifar-100 dataset are shown in Table 1. The proposed method favorably improves the performance of a vanilla KD [12] while outperforming the other KD methods. It should be noted that our extractive KD modifies only the teacher annotation in (9) without introducing additional matching measurement between student and teacher models such as in intermediate layers [24, 37]. The proposed annotation (9) explicitly introduces uniformity $\mathbf{u}$ as in the label smoothing regularization (LSR), thus being comparable to the performance of the LSR approach [27] which scratches the ground truth label by a small fraction of $\epsilon \in \{0.1, 0.2, 0.3\}$; the best performance over the parameter set is reported in Table 1. Our method is superior to the LSR, demonstrating that the uniformity $\mathbf{u}$ favorably works in the KD framework. While the ordinary KD implicitly utilizes uniformity embedded in the teacher probability $\mathbf{p}_\tau^t$ as analyzed in Sec. 3.3, we explicitly control it by a parameter ϵ to improve performance.

Then, our KD is compared with the sophisticated KD methods [29, 33, 31] which incorporate the other technique such as self-training [32] and effective data augmentation [35] into a KD framework. Our KD is competitive

Table 1. Performance results. For comparison, we apply LSR [27] and CutMix [35] to train student models without a help of teacher ones.

Teacher Student	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	ResNet56 ResNet20	ResNet110 ResNet20	ResNet32x4 ResNet8x4	VGG13 VGG8	VGG13 MobileV2	ResNet50 MobileV2	ResNet50 VGG8	ResNet32x4 ShuffleV2	WRN-40-2 ShuffleV1
Cifar-100 [16]											
Teacher	75.61	75.61	72.34	74.31	79.42	74.64	74.64	79.34	79.34	79.42	75.61
Student	73.26	71.98	69.06	69.06	72.50	70.36	64.60	64.60	70.36	71.82	70.50
KD [12]	74.92	73.54	70.66	70.67	73.33	72.98	67.37	67.35	73.81	74.45	74.83
FitNet [24]	73.58	72.24	69.21	68.99	73.50	71.02	64.14	63.16	70.69	73.54	73.73
AT [37]	74.08	72.77	70.55	70.22	73.44	71.43	59.40	58.58	71.84	72.73	73.32
PKT [22]	74.54	73.45	70.34	70.25	73.64	72.88	67.13	66.52	73.01	74.69	73.89
Ours	**75.36**	**74.26**	**71.17**	**71.29**	**75.55**	**73.89**	**69.41**	**69.39**	**73.93**	**76.17**	**75.30**
CRD [29]	75.48	74.14	71.16	71.46	75.51	73.94	69.73	69.11	74.30	75.65	76.05
SSKD [33]	**76.04**	**76.13**	71.49	71.08	76.20	**75.33**	71.53	**72.57**	75.76	78.61	77.40
KD+CutMix [31]	75.34	74.60	70.77	**71.82**	74.91	74.16	68.79	69.77	74.85	76.61	**77.63**
Ours+CutMix	**76.04**	75.52	**71.73**	71.15	**77.65**	74.55	**71.81**	72.13	**76.44**	**79.09**	77.62
LSR [27]	73.92	71.61	69.47	69.47	72.19	70.78	65.81	65.81	70.78	73.61	72.40
CutMix [35]	75.35	74.79	70.43	70.43	73.08	72.48	67.48	67.48	72.48	75.83	74.94
TinyImageNet [4]											
Teacher	61.87	61.87	58.29	59.85	64.50	62.52	62.52	69.39	69.39	64.50	61.87
Student	58.23	55.87	52.53	52.23	55.41	56.67	58.20	58.20	56.67	62.07	60.28
KD [12]	58.65	58.17	53.58	53.83	55.67	61.48	59.28	58.72	60.39	66.34	64.90
Ours	**60.22**	**60.30**	**54.43**	**54.91**	**59.87**	**62.51**	**62.07**	**62.29**	**61.59**	**67.17**	**65.19**
CRD [29]	60.79	59.31	**55.34**	55.17	59.28	62.92	62.38	61.56	62.03	67.33	65.44
SSKD [33]	59.73	59.18	53.50	54.12	57.73	62.95	62.39	62.79	63.18	67.27	64.39
KD+CutMix [31]	60.07	60.13	54.25	55.16	57.54	62.60	60.66	61.13	61.95	67.35	65.98
Ours+CutMix	**61.11**	**61.00**	54.67	**55.37**	**61.31**	**64.29**	**63.55**	**64.72**	**65.20**	**68.41**	**66.05**
LSR [27]	57.66	55.97	52.64	52.64	54.85	56.58	58.48	58.48	56.58	63.13	60.76
CutMix [35]	59.62	58.31	53.20	53.20	56.08	59.16	61.81	61.82	59.16	65.98	64.37

even with CRD [29] and it is straightforwardly extended in the CutMix framework [31] to produce competitive performance with those sophisticated methods. The combination method of our extractive KD and CutMix augmentation is also compared with the baseline method that trains a student model via CutMix and it is shown that our KD favorably works with the CutMix, which will be analyzed in Sec. 4.4.

In Table 1, the KD methods are also evaluated on Tiny-ImageNet dataset [4] which provides 200-category image classification task with 500 training image samples of 64×64 pixels per category as a subset of ImageNet dataset. As in Cifar-100, our method produces superior performance over the other methods and the combination method with CutMix favorably outperforms even the sophisticated KDs.

4.3. Feature transferability

The knowledge distillation is considered to enhance feature representation of the student model by reference to the teacher model. To assess the feature representation, we transfer pre-trained CNNs into the other task. In this experiment, the CNNs pre-trained by KD are frozen to provide features in a fixed way while only the (final) FC classifier layer is trained so as to be tuned toward the other classification tasks. For fairly evaluating feature representation, we apply normalized linear classification to produce logits $z = \frac{\boldsymbol{w}^\top \boldsymbol{x}}{\|\boldsymbol{w}\|\|\boldsymbol{x}\|}$ by means of L_2-normalization. In order to assess the generality of feature representation, we consider two types of transfer learning scenarios as follows.

One is to transfer CNNs pre-trained on Cifar-100 dataset into TinyImageNet task which forms homogeneous transfer as both two tasks belong to *object* classification. The other is the transfer from TinyImageNet to TinyPlaces365 [40] as heterogeneous transfer. Following Tiny-ImageNet, the Tiny-Places365 dataset is constructed by sampling 500 training images of 64×64 pixels per category from the Places365 dataset [40], rendering 365 *scene*-category classification in contrast to *object* classification in ImageNet. We apply two student CNN models of ResNet8x4 [9] and ShuffleNetV2 [18] both of which are pre-trained with the teacher ResNet32x4 [9] in the KD framework; for the performances on the primary tasks, refer to Table 1. The CNNs trained on 32×32 Cifar-100 images naturally cope with 64×64 images of TinyImageNet due to the global average pooling embedded in the ResNet and ShuffleNet which feeds a fixed-dimensional feature $\boldsymbol{x}$ to the last FC classifier.

The performance results are shown in Table 2. The proposed method produces competitive performance even to SSKD, and the combination with CutMix-training outperforms the others by a large margin. These experimental results on transfer learning demonstrate that the proposed method produces favorable feature representation. Though our method (without CutMix) is slightly inferior to SSKD

Table 2. Performance results on transfer learning.

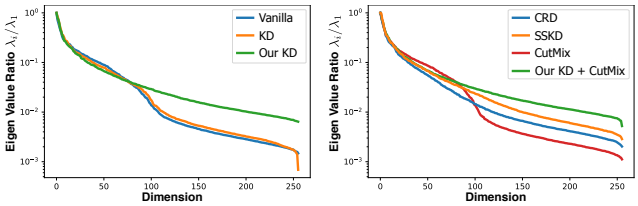
	Cifar100→TinyImageNet		TinyImageNet→TinyPlaces365	
Teacher	ResNet32x4		ResNet32x4	
Student	ResNet8x4	ShuffleV2	ResNet8x4	ShuffleV2
Teacher	26.36		21.52	
Student	30.76	30.73	25.06	22.03
KD	31.14	29.10	25.18	24.35
Ours	**36.27**	**36.67**	**27.48**	**27.47**
CRD	35.39	35.27	27.47	26.54
SSKD	36.63	36.56	26.59	27.33
KD+CutMix	33.43	31.09	25.99	25.76
Ours+CutMix	**37.61**	**39.70**	**28.33**	**29.76**
LSR	30.73	26.37	25.11	19.88
CutMix	29.73	27.95	25.04	22.40

in the primary task (Table 1), it works with slight superiority over SSKD in the transfer tasks. It implies that our KD effectively contributes to feature representation learning.

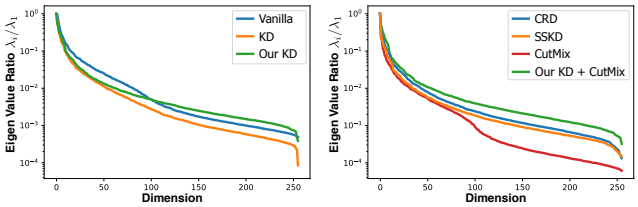
4.4. Analysis of feature representation

We finally analyze the feature representation learned by the proposed method. For that purpose, we measure the principal distribution of the features by means of eigenvalues in the PCA; $\sum_i (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\xi}_j = \lambda_j \boldsymbol{\xi}_j$, $j \in \{1, \cdots, d\}$ for features $\boldsymbol{x} \in \mathbb{R}^d$, and the eigenvalue indicates the feature variance along the corresponding principal direction (eigen vector). Fig. 9a shows the PCA eigenvalues by using 256-dimensional features of ResNet8x4 (student) trained with ResNet32x4 (teacher) on Cifar-100. The feature distribution of a vanilla CNN is concentrated in the 100-dimensional subspace since the eigenvalues λ_i of $i > 100$ are quite smaller than those of $i < 100$. The 100-category classification task of Cifar-100 would usually enforce features to lie in the 100 dimensional subspace; ideally speaking, $100 - 1 = 99$ dimensional simplex space is enough for discriminating 100 categories [5]. It, however, leads to over-fitting toward the Cifar-100 training samples, degrading generalization performance. The simple KD [12] exhibits similar tendency to the vanilla CNN. On the other hand, the proposed KD makes the features distribute rather *diversely* across dimensions to extract various features toward high generalization. The sophisticated KD methods of CRD [29] and SSKD [33] which explicitly incorporate self-training of feature representation learning also provide such a diverse feature distribution across dimensions. By CutMix training [35], the features are rather uniformly distributed *within* the 100-dimensional subspace due to mixing constraint [30], though providing less effective features beyond 100-dimensions. Thus, the combination of our KD and CutMix improves diversity across *all* the dimensions.

Fig. 9b shows the feature distribution on the *transfer learning* scheme (Sec. 4.3); we compute on TinyImageNet the eigenvalues of the ResNet8x4 features pre-trained on



(a) Cifar100 by ResNet8x4 pre-trained on Cifar100



(b) TinyImageNet by ResNet8x4 pre-trained on Cifar100

Figure 9. Distribution of PCA eigenvalues (feature variances).

Cifar-100; the feature extractor is the same as in Fig. 9a. The concentration on the *100*-dimensional subspace is also found in the vanilla and CutMix training, which implies the bias toward the primary task of Cifar-100 and thereby degrades performance on TinyImageNet of *200 category* classification. The proposed KD (+ CutMix) again exhibits diverse feature distributions even on the TinyImageNet dataset. Such a diverse feature representation favorably contributes to performance improvement (Table 2).

The standard KD based on a *raw* teacher probability is directly subject to the limitation of the teacher model which is highly biased toward the target dataset, e.g., Cifar-100 in Fig. 9. On the other hand, in the proposed KD, the teacher probability is transformed into the more effective form (9) through the extractive distillation process (Fig. 4). It encourages the student to extract diverse feature characteristics beyond not only the teacher model but also the ground truth label which only provides categorical information without its attributes regarding class relationships. The analysis of feature distribution in Fig. 9 clarifies that our extractive KD effectively trains CNNs to extract diverse features by suppressing bias toward the target dataset.

5. Conclusion

We have analyzed the process of knowledge distillation from various perspectives. We disentangled main KD components which mutually affect each other through a temperature and analyzed them separately to particularly clarify the effect of temperature. Through the analysis, an extractive distillation is proposed to further extract the effective knowledge from the teacher logits which is fed into KD formulation as a teacher annotation, The proposed KD method employs so-refined teacher knowledge, in contrast to other KDs which employ *raw* teacher probabilities. The experimental results on image classification tasks demonstrate the effectiveness of the proposed method in terms both of performance and feature representation.

References

[1] Atish Agarwala, Jeffrey Pennington, Yann Dauphin, and Sam Schoenholz. Temperature check: theory and practice for training models with softmax-cross-entropy losses. *arXiv preprint arXiv:2010.07344*, 2020.

[2] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *KDD*, pages 535–541, 2006.

[3] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *ICCV*, pages 4794–4802, 2019.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[5] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, Boston, 1990.

[6] Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-again neural networks. In *ICML*, pages 1607–1616, 2018.

[7] Vincent Gerbaud, Ivonne Rodriguez-Donis, Laszlo Hegely, Peter Lang, Ferenc Denes, and XinQiang You. Review of extractive distillation. process design, operation, optimization and control. *Chemical Engineering Research and Design*, 141:229–271, 2019.

[8] Jianping Gou, Baosheng Yu, Stephen J. Maybank, and acheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129:1789–1819, 2021.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[10] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *ICCV*, pages 1921–1930, 2019.

[11] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *AAAI*, pages 3779–3787, 2019.

[12] Geoffrey E. Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS 2014 Deep Learning Workshop*, 2014.

[13] Guangda Jin and Zhanxing Zhu. Knowledge distillation in wide neural networks: Risk bound, data efficiency and imperfect teacher. In *NeurIPS*, pages 20823–20833, 2020.

[14] Taehyeon Kim, Jaehoon Oh, Nak Yil Kim, Sangwook Cho, and Se-Young Yun. Comparing kullback-leibler divergence and mean squared error loss in knowledge distillation. In *IJCAI*, 2021.

[15] Takumi Kobayashi. Large margin in softmax cross-entropy loss. In *BMVC*, 2019.

[16] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[17] Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. In *NeurIPS*, pages 7517–7527, 2018.

[18] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, pages 116–131, 2018.

[19] Asit Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. In *ICLR*, 2018.

[20] Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019.

[21] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, pages 3967–3976, 2019.

[22] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *ECCV*, pages 268–284, 2018.

[23] Mary Phuong and Christoph H. Lampert. Towards understanding knowledge distillation. In *ICML*, pages 5142–5151, 2019.

[24] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, pages 535–541, 2015.

[25] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, and Jonathon Shlens. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[28] 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.

[29] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *ICLR*, 2020.

[30] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, pages 6438–6447, 2019.

[31] Huan Wang, Suhas Lohit, Michael Jones, and Yun Fu. Knowledge distillation thrives on data augmentation. *arXiv preprint arXiv:2012.02909*, 2020.

[32] Zhirong Wu, Yuanjun Xiong, X Yu Stella, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018.

[33] Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. In *ECCV*, 2020.

[34] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *CVPR*, pages 3903–3911, 2020.

[35] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019.

[36] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

[37] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[38] 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 *ICCV*, pages 3713–3722, 2019.

[39] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018.

[40] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018.