arXiv:2010.04812v1 [cs.LG] 9 Oct 2020

Local Region Knowledge Distillation

Xiang Deng, Zhongfei (Mark) Zhang
Computer Science, State University of New York at Binghamton

Abstract

Knowledge distillation (KD) is an effective technique to transfer knowledge from one neural network (teacher) to another (student), thus improving the performance of the student. The existing work trains the student to mimic the outputs of the teacher on training data. We argue that transferring knowledge at sparse training data points cannot enable the student to well capture the local shape of the teacher function. To address this issue, we propose locally linear region knowledge distillation (L^2RKD) which transfers the knowledge in local, liner regions from a teacher to a student. L^2RKD enforces the student to mimic the local shape of the teacher function in linear regions. Extensive experiments with various network architectures demonstrate that L^2RKD outperforms the state-of-the-art approaches by a large margin and is more data-efficient. Moreover, L^2RKD is compatible with the existing distillation methods and further improves their performances significantly.

Introduction

Deep neural networks (DNNs) have achieved state-of-the-art performances on a wide range of applications in artificial intelligence including computer vision (Krizhevsky, Sutskever, and Hinton 2012; Girshick 2015; He et al. 2016) and natural language processing (Andreas et al. 2016; Serban et al. 2016). The performance gain comes at a high cost of computation and memory in inference. This makes them unsuitable for deployment with limited resources or a strict latency requirement. One solution to this problem is knowledge distillation which transfers the knowledge from a large network (teacher) to a small and fast network (student). To the end, the student obtains a significant performance boost, even possibly matching or surpassing the performance of the teacher.

Hinton et al. (Hinton, Vinyals, and Dean 2015) proposed the original Knowledge Distillation (KD) that utilizes the soft labels generated by a teacher as the targets to train a student. From the perspective of function curve fitting, the student function is to learn the local surface shape of the teacher function at each training data point. It is widely believed that the generalization ability of a DNN is highly related to the stability which can be described as the local function shape of a DNN. When we have a large teacher network which typically has a much better generalization performance than that of the student, we can suppose that the teacher function has a more stable (better) shape within the data distribution than that of the student. Intuitively, if the student better captures the local shape of the teacher, it can have a better generalization ability. However, the existing work only transfers the knowledge at each individual training data point from a teacher to a student as the real data distribution is typically unknown. Sparse individual data point fitting may not necessarily enable the student to well capture the local shape of the teacher function. As shown in Figure 1, even if the student f_S perfectly fits the teacher f_T at each training data point, i.e., x_1, x_2, and x_3, their local shapes near these data points can be highly different.

To alleviate this problem, the typically used strategy is to train on different but similar data points to the training data. This is known as data augmentation (Simard et al. 1998) formalized by the *Vicinal Risk Minimization* (VRM) (Chapelle et al. 2001) principle. In VRM, human knowledge is necessary to define a vicinity or neighborhood around each training data point. Then, additional new data points can be drawn from the vicinity distribution of the training data to enlarge the support of the training data distribution. For example, in image classification, it is common to define the vicinity of an image as the set of its random flippings and crops after a mildly padding. Nevertheless, data augmentation has its own limitation that a new generated data point is very close to the original training data point, since they contain almost the identical objective only with different backgrounds caused by padding or cropping. Due to this limitation, as shown in Figure 2, even if the student f_S fits the teacher f_T at all the training data points (i.e., x_1, x_2, and x_3) and the augmented data points (i.e., x_1^{aug}, x_2^{aug}, and x_3^{aug}), their local shapes can still differ substantially.

To address the above issue, we propose a regularizer for KD to enforce the student to mimic the shape of the teacher in local, linear regions, thus achieving locally linear region knowledge distillation (L^2RKD). As shown in Figure 3, instead of only fitting the data points near the training samples, L^2RKD also fits the data points in the linear region between two augmented samples, e.g., $\hat{x}_1$ and $\hat{x}_2$ in the linear region between x_1^{aug} and x_2^{aug}, and $\hat{x}_3$ in the linear region between x_2^{aug} and x_3^{aug}. The data points in a linear region can be represented by a linear combination of two augmented examples, i.e., $\gamma * x_i + (1-\gamma) * x_j$ where x_i and x_j are two samples

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

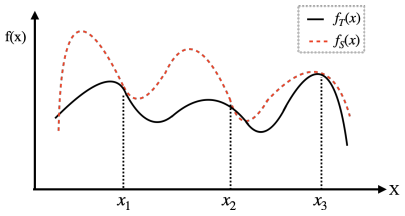


Figure 1: KD without data augmentation

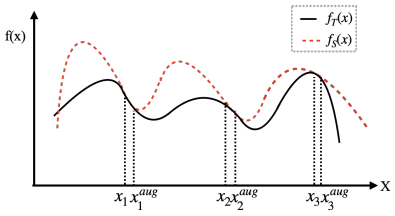


Figure 2: KD with data augmentation

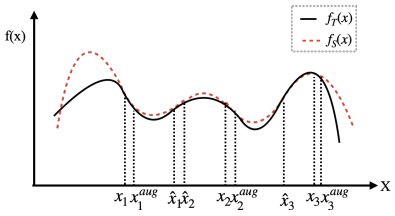


Figure 3: L^2RKD

drawn from the vicinity distribution of the training data and γ is a random variable within range [0, 1]. As seen from Figure 3, distilling knowledge in local, linear regions can make the student better capture the local shape of the teacher.

We summarize the main contributions as follows:

- We observe that only fitting the data points in the vicinity around training data cannot enable a student to well capture the local shape of a teacher. To address this issue, we propose L^2RKD which transfers the knowledge in local, linear regions from a teacher to a student.
- We compare L^2RKD with more than 10 state-of-the-art approaches. Despite its simplicity, L^2RKD outperforms these methods substantially and is more data-efficient.
- L^2RKD is compatible with the existing distillation methods to further improve their performances significantly.

Related Work

Existing knowledge distillation approaches can be mainly divided into two categories, i.e., logit-based approaches and feature-based approaches. Their objectives can be simply represented as a combination of a regular cross entropy objective and a distillation objective. Logit-based approaches construct the distillation objective based on output logits. Hinton et al. (Hinton, Vinyals, and Dean 2015) propose KD which uses the softened logits of a teacher network as the targets to train a student network. Park et al. (Park et al. 2019) propose to transfer mutual relations of data examples from a teacher to a student by penalizing logit-based structural differences between them. On the other hand, feature-based approaches construct the distillation objective based on feature maps in intermediate or last several layers. FitNet (Romero et al. 2015) uses the features of a teacher as hints to train a student through regressions. Zagoruyko and Komodakis (Zagoruyko and Komodakis 2017) propose to transfer attention from a teacher to a student. The attention is obtained from feature maps. CRD (Tian, Krishnan, and Isola 2020) proposes to maximize the mutual information between student and teacher representations. Other feature-based methods (Yim et al. 2017; Huang and Wang 2017; Kim, Park, and Kwak 2018; Liu et al. 2019; Srinivas and Fleuret 2018; Wang et al. 2018; Heo et al. 2019a; Cho and Hariharan 2019; Ahn et al. 2019; Koratana et al. 2019; Aguilar et al. 2019) deal with feature representations utilizing different criteria. Online knowledge distillation (Zhang et al. 2018b; Chen et al. 2020; Anil et al. 2018; Chung et al. 2020; Zhu, Gong et al. 2018) has also been proposed to train a group of students simultaneously.

The local, linear regions in L^2RKD are obtained by linearly combining samples. Linear interpolation has been used in many applications with different goals and mechanisms. $Mixup$ (Zhang et al. 2018a) enforces local linearity of a DNN by linearly interpolating a random pair of training samples and their one-hot labels simultaneously. However, simply interpolating two labels may not match the generated sample as pointed out in (Guo, Mao, and Zhang 2019). For example, as shown Figure 4, (c) is a linearly interpolated image from (a) and (b). The interpolated label for (c) has high probabilities at 1 and 5. However, the real label of (c) should have a high probability at 8. This kind of unmatched image and label pairs may cause undesirable results for training DNNs. L^2RKD does not have the above issue as it takes advantages of the knowledge in a powerful teacher. L^2RKD aims to teach a student to mimic the local shape of a powerful teacher while Mixup aims to enforce local linearity of a DNN function. In addition, the interpolation technique has also been used in semi-supervised learning (Berthelot et al. 2019) and also used to improve the robustness (Li, Socher, and Hoi 2020), uncertainty (Hendrycks et al. 2019), and calibration (Thulasidasan et al. 2019) of DNN classifiers.



(a) (b) (c)

Figure 4: Mixup

Our Framework

From KD to L^2RKD

Hinton et al. (Hinton, Vinyals, and Dean 2015) propose KD which minimizes the output probability differences between a student and a teacher over data distribution $P(X, Y)$. The complete objective is:

$$\mathcal{L}_{KD} = \mathop{\mathbb{E}}_{P(X,Y)}[\alpha \mathcal{L}_{CE}(x, y) + (1 - \alpha)\mathcal{L}_{KL}(x)] \quad (1)$$

where α is a balancing weight and $\mathcal{L}_{CE}$ is the regular cross-entropy objective:

$$\mathcal{L}_{CE}(x, y) = H(y, \sigma(f_S(x))) \quad (2)$$

where $H(.)$ is the cross-entropy loss and σ is softmax. $\mathcal{L}_{KL}(x)$ in (1) is the distillation objective:

$$\mathcal{L}_{KL}(x) = \tau^2 KL\left(\sigma\left(\frac{f_T(x)}{\tau}\right), \sigma\left(\frac{f_S(x)}{\tau}\right)\right) \quad (3)$$

where τ is a temperature to generate soft labels and KL represents KL-divergence. The gradient of $\mathcal{L}_{KL}$ with respect

to the logits $f_S(x)$ is written as:

$$\frac{\partial \mathcal{L}_{KL}(x)}{\partial f_S(x)} = \tau \left(\sigma\left(\frac{f_S(x)}{\tau}\right) - \sigma\left(\frac{f_T(x)}{\tau}\right)\right) \quad (4)$$

When τ is large compared with the magnitude of the logits, the gradient can be approximated (based on Taylor series) as:

$$\frac{\partial \mathcal{L}_{KL}(x)}{\partial f_S(x)} \approx \tau \left(\frac{1+\frac{f_S(x)}{\tau}}{K+\sum_j \frac{f_S(x)[j]}{\tau}} - \frac{1+\frac{f_T(x)}{\tau}}{K+\sum_j \frac{f_T(x)[j]}{\tau}} \right) \quad (5)$$

where K is the total number of classes and $f_S(x)[j]$ represents the jth element of $f_S(x)$. With the assumption that the logits of a DNN have mean zero, we can obtain:

$$\frac{\partial \mathcal{L}_{KL}(x)}{\partial f_S(x)} \approx \frac{1}{K}\left(f_S(x) - f_T(x)\right) \quad (6)$$

As seen from the gradient, minimizing $\mathcal{L}_{KL}(x)$ with a high temperature is equivalent to minimizing the mean square error between $f_S(x)$ and $f_T(x)$. Thus, the KD process can be considered as using one function (student) to fit another fixed function (teacher) over the data distribution $P(X,Y)$. However, the data distribution $P(X,Y)$ is typically unknown in practical situations. Instead, we usually have a set of training data $\mathcal{T}$=$\{(x_1,y_1), (x_2,y_2), ..., (x_n,y_n)\}$ where $(x_i,y_i) \sim P(X,Y)$. With training data $\mathcal{T}$, P can be approximated by the *empirical distribution*:

$$P_\delta(x,y) = \frac{1}{n}\sum_{i=1}^{n} \delta(x=x_i, y=y_j) \quad (7)$$

where $\delta(x=x_i, y=y_j)$ is a Dirac mass centered at (x_i,y_i). Then loss (1) is approximated by the *empirical risk*:

$$\begin{aligned}\mathcal{L}_\delta &= \mathop{\mathbb{E}}_{P_\delta(X,Y)}[\alpha\mathcal{L}_{CE}(x,y) + (1-\alpha)\mathcal{L}_{KL}(x)] \\ &= \frac{1}{n}\sum_{i=1}^{n}(\alpha\mathcal{L}_{CE}(x_i,y_i) + (1-\alpha)\mathcal{L}_{KL}(x_i))\end{aligned} \quad (8)$$

Training f_S by minimizing (8) is known as the *Empirical Risk Minimization* (ERM) principle (Vapnik 1998). While efficient to compute, the empirical risk (8) only guides student f_S to fit teacher f_T at a finite set of sparse training data points. As shown in Figure 1, this may not enable the student to well capture the local shape of the teacher which plays an important role on the generalization ability.

To alleviate this issue, data augmentation based on the VRM principle is typically used. In VRM, the data distribution P is approximated by a human defined vicinity distribution of the training data $\mathcal{T}$:

$$P_v(x^{aug}, y^{aug}) = \frac{1}{n}\sum_{i=1}^{n} v(x^{aug}, y^{aug}|x_i, y_i) \quad (9)$$

where v is a vicinity distribution that measures the probability of (x^{aug}, y^{aug}) existing in the neighborhood of training data point (x_i,y_i). In practice, we usually only augment each training sample x_i without changing label y_i. For instance, in image classification, x_i^{aug} can be obtained by slightly padding and randomly cropping x_i and remaining label y_i. We use VRM by drawing samples from $P_v(x^{aug}, y^{aug})$ to obtain a new training dataset $\mathcal{T}_v$=$\{(x_1^{aug1}, y_1)$,(x_1^{aug2}, y_1), ..., (x_1^{augm}, y_1), ..., (x_n^{aug1}, y_n), (x_n^{aug2}, y_n), ..., $(x_n^{augm}, y_n)\}$ where (x_i^{aug1}, y_i), (x_i^{aug2}, y_i), ..., (x_i^{augm}, y_i) are m samples drawn from the vicinity of (x_i,y_i) (m can be considered as the total training epochs in SGD optimization). This leads to the *empirical vicinal risk*:

$$\begin{aligned}\mathcal{L}_v = \frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}[&\alpha\mathcal{L}_{CE}(x_i^{augj}, y_i) \\ &+(1-\alpha)\mathcal{L}_{KL}(x_i^{augj})]\end{aligned} \quad (10)$$

As shown in Figure 2, with data augmentation, the student can better capture the local shape of the teacher, but still only covers a very small region around each training sample x_i.

To address this issue, we propose a data-driven regularizer for KD to transfer the knowledge in local, linear regions from the teacher to the student. The distribution of the data points in linear regions is defined as:

$$\begin{aligned}P_{lr}(\hat{x}) = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}_{\gamma\sim B(b,b)}[\\ \delta(\hat{x} = \gamma * x_i^{aug} + (1-\gamma) * x_j^{aug})]\end{aligned} \quad (11)$$

where $B(b,b)$ is the Beta distribution $Beta(b,b)$ with hyperparameter b. In a nutshell, $\hat{x}$ is a randomly and linearly interpolated data point from two samples (e.g., x_i^{aug} and x_j^{aug}) in the vicinity around training data: $\hat{x} = \gamma * x_i^{aug} + (1-\gamma) * x_j^{aug}$. Note that the idea can be easily extended to a linear combination of three or more samples with weights γ drawn from a Dirichlet distribution.

We develop the following regularizer to enforce the student to mimic the behavior of the teacher in linear regions:

$$\mathcal{L}_{reg} = \mathop{\mathbb{E}}_{P_{lr}(\hat{x})} \tau^2 KL\left(\sigma\left(\frac{f_T(\hat{x})}{\tau}\right), \sigma\left(\frac{f_S(\hat{x})}{\tau}\right)\right) \quad (12)$$

Samples can be drawn from $P_{lr}(\hat{x})$, which leads to L^2RKD objective $\mathcal{L}_{lr}$:

$$\begin{aligned}\frac{1}{mn}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(\alpha\mathcal{L}_{CE}(x_i^{augj}, y_i) + (1-\alpha)\mathcal{L}_{KL}(x_i^{augj})\right) \\ +\eta\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathcal{L}_{KL}(\hat{x}_{ij}))\end{aligned} \quad (13)$$

where $\hat{x}_{ij}$ is a sample drawn from $P_{lr}(\hat{x})$ and η is a balancing weight. The first term of (13) enables the student to fit the local shape of the teacher in the vicinity around each training data point and the second term makes the student fit the local shape of the teacher in each linear region. Note that when γ equals to 0 or 1, $\gamma * x_i^{aug} + (1-\gamma) * x_j^{aug}$ results in the data points in the vicinity of x_j or x_i. Thus, (13) pays more attention to the samples in the vicinity of training data. This is reasonable as these points are closer to the real data distribution than those in linear regions. The implementation-level description of L^2RKD is presented in Algorithm 1.

Algorithm 1 L^2RKD

Input: Training data $\mathcal{T}$, teacher f_T, and student f_S
Output: Student f_S with optimal parameters

for $i = 1, 2, ..., k$ **do**
  Randomly take a mini-batch of training data (x_{batch}, y_{batch}) with mini-batch size $\mathcal{S}$ from $\mathcal{T}$
  Generate x_{batch}^{aug} by using data augmentation (e.g., randomly flipping, padding, and cropping)
  Randomly shuffle x_{batch}^{aug} to obtain $x_{batch}^{aug'}$
  Sample weights $\gamma \sim Beta(b, b)$
  Obtain $\hat{x}_{batch} = \gamma * x_{batch}^{aug} + (1 - \gamma) * x_{batch}^{aug'}$
  Compute the training loss of L^2RKD with $\mathcal{L}_{lr} = \frac{1}{\mathcal{S}} (\alpha \mathcal{L}_{CE}(x_{batch}^{aug}, y_{batch}) + (1 - \alpha)\mathcal{L}_{KL}(x_{batch}^{aug})) + \frac{1}{\mathcal{S}} \eta \mathcal{L}_{KL} (\hat{x}_{batch})$
  Minimize $\mathcal{L}_{lr}$ with one gradient descent step for f_S
end for

Analysis of L^2RKD

L^2RKD distills the knowledge in local, linear regions instead of sparse data points. This is achieved by using a data-driven regularizer to require the student to mimic the shape of the teacher in local, linear regions as shown in Figure 3. One may consider this data-driven regularizer as a data augmentation strategy, but there are many worth-noting differences. Regular data augmentation usually generates in-distribution samples with ground truth, i.e., (x, y) where (x, y) follows the real data distribution $P(X, Y)$. These generated pairs are used to learn an ideal function f mapping x to y where $(x, y) \sim P(X, Y)$. In contrast, the proposed regularizer aims to enforce a student f_S to mimic the shape of a teacher f_T in local, linear regions. The local, linear regions contain in-distribution samples **without ground truth**, near-distribution samples, and far-out-of-distribution samples. These samples assist a student f_S to learn the distribution $Q(X, Z)$ defined by teacher f_T where X are the inputs and Z are the output probabilities of f_T. Note that $Q(X, Z)$ is different from $P(X, Y)$ as f_t is usually not perfect. The advantage of fitting $Q(X, Z)$ is that $Q(X, Z)$ **is more tractable than** $P(X, Y)$, since for given any sample x, f_T can always give output $f_T(x)$ and $(x, \sigma(\frac{f_T(x)}{\tau}))$ follows $Q(X, Z)$. Even for out-of-distribution samples in the linear regions, f_T can still output soft labels. It is obvious that these soft labels for out-of-distribution samples are semantically meaningless. However, by querying f_T in local, linear regions, the student is able to better capture the local shape of the teacher as shown in Figure 3, thus obtaining more performance gain.

Experiments

Datasets and Architectures

We report the results on four benchmark datasets, i.e., CIFAR-10 (Krizhevsky and Hinton 2009), CIFAR-100, Tiny ImageNet [1], and ImageNet (Deng et al. 2009). CIFAR-10 is a 10-class image classification dataset, containing 50,000 training images and 10,000 test images. CIFAR-100 has similar images to those in CIFAR-10, but has 100 classes. Tiny ImageNet has 200 classes, containing 100,000 training images and 10,000 test images. ImageNet is a large-scale image classification dataset with 1000 classes, containing 1.28 million training images and 50,000 validation images. We use the standard data augmentation strategy for each dataset.

To evaluate the applicability of L^2RKD on different network architectures, we adopt various modern architectures, i.e., ResNet (He et al. 2016), WRN (Zagoruyko and Komodakis 2016), VGG (Simonyan and Zisserman 2015), MobileNet (Sandler et al. 2018), and ShuffleNet (Ma et al. 2018).

Experimental Setup

We compare L^2RKD with several state-of-the-art methods, i.e., KD (Hinton, Vinyals, and Dean 2015), FitNet (Romero et al. 2015), AT (Zagoruyko and Komodakis 2017), SP (Tung and Mori 2019), FT (Kim, Park, and Kwak 2018), NST (Huang and Wang 2017), CC (Peng et al. 2019), FSP (Yim et al. 2017), PKT (Passalis and Tefas 2018), AB (Heo et al. 2019b), VID (Ahn et al. 2019), RKD (Park et al. 2019), and CRD (Tian, Krishnan, and Isola 2020).

For L^2RKD, α and η are set to 0.1 and 1, respectively. b in Beta distribution is set to 1 (generally, $b \in [1, 3]$ works very well) for all the datasets except ImageNet (0.2). All networks are trained with SGD with momentum 0.9 for 240 epochs (100 epochs on Tiny ImageNet). For the state-of-the-art approaches, we report the author-reported results, or use author-provided codes and the optimal hyper-parameters from the original papers if they are publicly available. Otherwise, we use the implementation of (Tian, Krishnan, and Isola 2020). All the results below are reported based on 3 runs. More implementation details are reported in the Appendix.

Comparison with State-of-the-art Methods

Table 1 summarizes the comparison results on CIFAR-10. We have the following observations. First, by imposing the simple regularizer $\mathcal{L}_{reg}$, L^2RKD consistently outperforms KD and the other state-of-the-art approaches significantly across different architectures, which demonstrates the effectiveness of L^2RKD and indicates the importance of the knowledge in local, linear regions. Second, the approaches based on logits or the last several layers' features (e.g., KD, PKT, CRD, and L^2RKD) perform better than the approaches based on intermediate features (e.g., FitNet and AT). The reason can be that the output logits and the last several layers' features of the teacher contain more high-level information that the features in the shallow layers. Third, when the teacher architecture and the student architecture are extremely different (e.g., VGG-13 and MobileNetV2), almost all feature-based approaches (e.g., FitNet and AT) fail to transfer knowledge from the teacher to the student, even underperforming the vanilla student. This indicates that aligning the features of a teacher and a student with extremely different architectures hurts the student performance. In contrast, the approaches based on soft labels obtained from logits (e.g., KD and L^2RKD) can still improve the student performance.

We further evaluate L^2RKD on different datasets. Table 2 and Table 3 report the comparison results on CIFAR-100 and Tiny ImageNet, respectively. It is observed that L^2RKD

[1] https://tiny-imagenet.herokuapp.com

Table 1: Test accuracies (%) on CIFAR-10

Teacher Student	WRN-40-2 WRN-16-1	ResNet32×4 ResNet8×4	WRN-16-4 WRN-16-1	ResNet-110 MobileNetV2	VGG-13 MobileNetV2	ResNet-110 WRN-16-1
Teacher	94.80	95.52	95.04	94.00	94.01	94.00
Vanilla Student	91.32	92.78	91.32	89.19	89.19	91.32
KD	91.77±0.25	93.85±0.22	92.62±0.20	89.61±0.26	89.21±0.17	91.52±0.16
FitNet	91.20±0.12	93.21±0.29	90.77±0.34	88.84±0.25	87.51±0.21	91.15±0.15
AT	91.96±0.29	93.45±0.19	91.88±0.10	89.40±0.33	87.06±0.09	91.41±0.26
SP	91.78±0.18	93.31±0.15	92.19±0.31	89.09±0.36	88.55±0.28	91.07±0.27
CC	91.00±0.26	92.95±0.32	91.01±0.35	88.13±0.41	87.34±0.15	91.04±0.10
VID	91.47±0.20	93.65±0.25	90.99±0.11	89.50±0.22	87.23±0.25	91.56±0.28
RKD	91.50±0.25	93.20±0.19	92.25±0.31	89.20±0.17	88.45±0.27	91.32±0.12
AB	91.01±0.18	92.88±0.28	90.70±0.51	88.37±0.19	87.81±0.29	90.52±0.26
PKT	91.97±0.17	94.09±0.27	92.28±0.08	89.18±0.15	88.60±0.12	91.91±0.14
CRD	91.95±0.28	94.08±0.17	89.67±0.48	85.70±0.10	89.16±0.12	91.80±0.21
FT	91.01±0.25	92.97±0.15	91.30±0.09	80.56±0.41	83.44±0.19	90.92±0.05
NST	91.56±0.11	92.92±0.20	91.14±0.24	88.77±0.34	82.95±0.42	90.66±0.15
L^2RKD	**93.11±0.21**	**94.75±0.19**	**93.24±0.25**	**90.83±0.18**	**90.52±0.11**	**92.47±0.07**

Table 2: Test accuracies (%) on CIFAR-100

Teacher Student	VGG-13 VGG-8	ResNet32×4 ResNet8×4	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	ResNet-56 ResNet-20	ResNet-110 ResNet-32
Teacher	74.64	79.42	75.61	75.61	72.34	74.31
Vanilla Student	70.36	72.50	73.26	71.98	69.06	71.14
KD	72.98±0.19	73.33±0.25	74.92±0.28	73.54±0.20	70.66±0.24	73.08±0.18
FitNet	71.02±0.31	73.50±0.28	73.58±0.32	72.24±0.24	69.21±0.36	71.06±0.13
AT	71.43±0.09	73.44±0.19	74.08±0.25	72.77±0.10	70.55±0.27	72.31±0.08
SP	72.68±0.19	72.94±0.23	73.83±0.12	72.43±0.27	69.67±0.20	72.69±0.41
CC	70.71±0.24	72.97±0.17	73.56±0.26	72.21±0.25	69.63±0.32	71.48±0.21
VID	71.23±0.06	73.09±0.21	74.11±0.24	73.30±0.13	70.38±0.14	72.61±0.28
RKD	71.48±0.05	71.90±0.11	73.35±0.09	72.22±0.20	69.61±0.06	71.82±0.34
PKT	72.88±0.09	73.64±0.18	74.54±0.04	73.45±0.19	70.34±0.04	72.61±0.17
AB	70.94±0.18	73.17±0.31	72.50±0.26	72.38±0.31	69.47±0.09	70.98±0.39
FT	70.58±0.08	72.86±0.12	73.25±0.20	71.59±0.15	69.84±0.12	72.37±0.31
FSP	70.23±0.23	72.62±0.13	72.91±0.24	n/a	69.95±0.21	71.89±0.11
NST	71.53±0.13	73.30±0.25	73.68±0.11	72.24±0.22	69.60±0.13	71.96±0.07
L^2RKD	**74.54±0.15**	**75.67±0.18**	**75.91±0.22**	**75.34±0.11**	**71.30±0.19**	**73.61±0.21**

shows consistent improvements over all the other approaches on both datasets across different architectures, which demonstrates the superiority of L^2RKD across datasets and architectures.

To further explore the performance of L^2RKD on large-scale datasets, we report the comparison results on ImageNet. We follow CRD by using ResNet-34 and ResNet-18 as the teacher and the student networks, respectively. As shown in Table 4, L^2RKD beats the other approaches significantly, which demonstrates the usefulness and applicability of L^2RKD on large scale datasets.

Compatibility with State-of-the-art Approaches

Feature-based approaches can be combined with logit-based approaches to further improve their performances. Currently, state-of-the-art approaches are combined with KD to obtain further performance gain. Here, we show that the state-of-the-art approaches combined with L^2RKD are able to obtain more performance gain. As shown in Table 5, the existing approaches combined with L^2RKD consistently outperform themselves combined with KD on all the settings where the teachers and students share similar architectures (e.g., WRN-40-2 and WRN-16-2) or different architectures (e.g., ResNet50 and VGG-8). This demonstrates that L^2RKD is compatible with the existing approaches and is able to further improve their performances significantly.

Data Efficiency

In many practical situations, a powerful model is released, but only a few data samples are publicly accessible due to the privacy or confidentiality issues in various domains such as medical and industrial domains. It is essential for knowledge

Table 3: Test accuracies (%) on Tiny ImageNet

Teacher	Student	KD	FitNet	AT	CC	SP	VID	RKD	CRD	PKT	AB	L^{2}RKD
VGG-13 (61.62)	VGG-8 (55.46)	60.21 ±0.19	55.26 ±0.20	56.82 ±0.46	54.14 ±0.19	56.99 ±0.42	54.57 ±0.26	56.60 ±0.13	59.95 ±0.23	56.36 ±0.17	55.41 ±0.36	**62.31 ±0.16**
WRN-40-2 (61.84)	WRN-40-1 (55.39)	56.25 ±0.15	55.41 ±0.31	55.84 ±0.41	55.10 ±0.43	54.09 ±0.26	56.07 ±0.23	55.37 ±0.29	56.75 ±0.33	56.31 ±0.22	55.76 ±0.26	**57.60 ±0.19**

Table 4: Comparison results on ImageNet

	Teacher	Student	KD	AT	CRD	SP	CC	L^{2}RKD
TOP-1	73.31	69.75	70.66	70.70	71.17	70.22	69.96	**71.48**
TOP-5	91.42	89.07	89.88	90.00	90.13	89.80	89.17	**90.15**

Table 5: Compatibility performances on CIFAR-100

Teacher Student	ResNet32×4 ResNet8×4	VGG-13 VGG-8	WRN-40-2 WRN-16-2	ResNet-50 VGG-8	ResNet32×4 ShuffleNetV2
Teacher	79.42	74.64	75.61	79.34	79.42
Vanilla Student	72.50	70.36	73.26	70.36	71.82
FitNet+KD	74.66	73.22	75.12	73.24	75.15
FitNet+L^{2}RKD	**76.36**	**74.73**	**75.81**	**75.02**	**77.36**
AT+KD	74.53	73.48	75.32	74.01	75.39
AT+L^{2}RKD	**75.47**	**74.11**	**75.81**	**75.16**	**76.22**
SP+KD	74.02	73.49	74.98	73.52	74.88
SP+L^{2}RKD	**75.35**	**73.80**	**75.05**	**74.81**	**75.57**
CC+KD	74.21	73.04	75.09	73.48	74.71
CC+L^{2}RKD	**75.64**	**74.69**	**75.69**	**75.17**	**76.29**
VID+KD	74.56	73.19	75.14	73.46	74.85
VID+L^{2}RKD	**75.89**	**74.87**	**75.68**	**75.15**	**76.55**
RKD+KD	73.79	72.97	74.89	73.51	74.55
RKD+L^{2}RKD	**74.96**	**74.58**	**75.65**	**74.75**	**76.46**
PKT+KD	74.23	73.25	75.33	73.61	74.66
PKT+L^{2}RKD	**75.54**	**74.28**	**75.74**	**75.42**	**76.59**
CRD+KD	75.46	74.29	75.64	74.58	76.05
CRD+L^{2}RKD	**76.51**	**74.69**	**76.16**	**75.05**	**76.83**
AB+KD	74.40	73.35	70.27	73.65	74.99
AB+L^{2}RKD	**76.30**	**74.67**	**71.51**	**75.24**	**77.73**
NST+KD	74.28	73.33	74.67	71.74	75.24
NST+L^{2}RKD	**75.12**	**74.19**	**75.47**	**72.81**	**76.73**

Table 6: S-T DIFs (shape differences) on CIFAR-100

Teacher Student	ResNet32×4 ResNet8×4	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	VGG-13 VGG-8
KD	2.81	2.74	2.94	1.77
L^{2}RKD	**1.59**	**2.11**	**2.21**	**1.25**



Figure 5: Effects of b

distillation methods to work in these data-limited cases. We evaluate L^{2}RKD in the experimental setting where knowledge is transferred from a powerful teacher to a student with limited data available. Table 7 presents the comparison results. We observe that L^{2}RKD substantially outperforms the state-

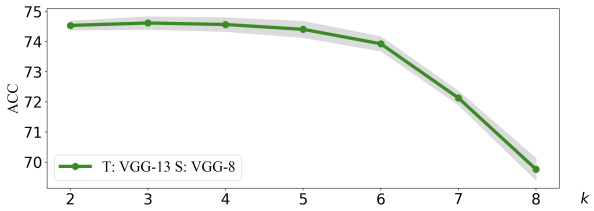


Figure 6: Effects of k

of-the-art approaches in all the cases with 60%, 40%, 20%, and 10% training data available, which indicates that L^{2}RKD is more data-efficient and demonstrates the usefulness of the knowledge in local, linear regions on data-limited situations. We also evaluate the compatibility of L^{2}RKD with the existing approaches in data-limited situations. As shown in Table 7, these approaches achieve much better performances when combined with L^{2}RKD than when combined with KD, which demonstrates the superior compatibility of L^{2}RKD with the existing methods in the data-limited cases.

Shape Differences

In this part, we verify that the students trained by L^{2}RKD better capture the local shapes of the teachers than those trained by KD. The local shape of a function can be represented by a set of pairs (x, y) where x is the input and y is the output of the function. To measure the shape difference, we report the average mean square **student-teacher output logit differences (S-T DIFs)** by using test data as inputs. As shown in Table 6, S-T DIFs of L^{2}RKD are consistently smaller than those of KD, which demonstrates that the student shapes of L^{2}RKD are closer to the teacher shapes and indicates that the knowledge in locally linear regions is beneficial for the students to capture the local shapes of the teachers.

Effects of b in Beta Distribution

b in $Beta(b, b)$ controls the interpolation extent between data point pairs. Setting b to 1 leads to interpolations uniformly distributed between 0 and 1 while setting it to 0 results in the empirical vicinal risk. Figure 5 shows how the performance varies with the values of b on CIFAR-100. We note that $b \in [1.0, 3.0]$ works very well, which indicates that L^{2}RKD is not sensitive to b.

Effects of Sample Number in Linear Combination

L^{2}RKD can be easily extended to a randomly linear combination of more than two samples, which results in $\hat{x} = w_1 * x_1^{aug} + w_2 * x_2^{aug} +, ..., +w_k * x_k^{aug}$ where $k > 2$, w = $[w_1, w_2, ..., w_k]$, and $w \sim Dirichlet(b, b, b, ..., b)$ (totally k bs). Figure 6 shows the effect of k on CIFAR-100. It is

Table 7: Data efficiency performances on CIFAR-100

	60% Training Data		40% Training Data		20% Training Data		10% Training Data	
Teacher	ResNet32×4	VGG-13	ResNet32×4	VGG-13	ResNet32×4	VGG-13	ResNet32×4	VGG-13
Student	ResNet8×4	VGG-8	ResNet8×4	VGG-8	ResNet8×4	VGG-8	ResNet8×4	VGG-8
Teacher	79.42	74.64	79.42	74.64	79.42	74.64	79.42	74.64
Vanilla Student	68.54	65.57	64.35	61.45	54.70	52.50	42.76	39.30
KD	69.12	69.90	66.44	66.89	58.23	59.14	47.95	49.00
FitNet	69.61	66.60	66.97	62.06	60.18	53.57	51.46	39.89
AT	69.35	67.36	66.19	65.12	57.72	58.16	44.70	47.16
SP	69.62	69.76	65.82	66.40	59.00	58.57	45.44	40.27
CC	68.37	65.37	64.26	60.60	54.68	51.27	42.74	39.16
VID	69.12	67.29	65.87	62.58	57.31	54.86	44.41	41.07
RKD	67.71	66.18	63.51	62.32	52.29	52.13	39.19	39.70
PKT	70.48	69.21	66.41	65.97	59.06	58.08	43.50	41.15
CRD	71.29	70.46	68.15	66.27	59.38	57.57	48.23	46.33
AB	69.25	65.98	65.30	63.07	58.48	56.55	48.61	48.27
FT	67.05	64.88	63.38	60.37	53.85	50.42	39.55	38.10
NST	69.87	67.12	66.24	63.56	60.27	56.63	51.91	47.44
L^2RKD	**72.31**	**71.62**	**69.55**	**69.31**	**63.11**	**63.95**	**54.56**	**54.85**
FitNet+KD	72.09	69.21	69.64	65.94	63.36	58.41	54.72	46.30
FitNet+L^2RKD	**74.03**	**71.87**	**71.80**	**69.68**	**66.58**	**63.57**	**58.79**	**50.56**
AT+KD	71.54	70.61	68.01	68.22	60.89	62.78	50.22	53.91
AT+L^2RKD	**72.59**	**72.20**	**70.17**	**69.60**	**64.21**	**64.50**	**55.60**	**55.89**
SP+KD	70.45	69.70	67.22	66.73	60.53	60.35	49.94	45.62
SP+L^2RKD	**72.41**	**71.67**	**69.33**	**69.65**	**63.90**	**63.56**	**54.41**	**50.97**
CC+KD	70.67	69.38	66.81	66.27	59.03	58.60	48.54	48.37
CC+L^2RKD	**72.36**	**71.66**	**69.40**	**69.35**	**63.50**	**63.33**	**53.51**	**53.15**
VID+KD	70.00	69.54	67.65	66.42	60.03	58.68	47.84	46.81
VID+L^2RKD	**72.75**	**72.41**	**70.23**	**69.75**	**64.49**	**64.18**	**54.56**	**54.54**
RKD+KD	70.33	69.74	66.63	66.15	58.54	58.95	46.43	48.20
RKD+L^2RKD	**72.16**	**71.69**	**69.24**	**69.60**	**63.08**	**63.28**	**51.77**	**56.62**
PKT+KD	70.98	70.18	67.43	66.13	60.26	59.48	48.78	47.88
PKT+L^2RKD	**72.43**	**72.21**	**69.81**	**69.51**	**63.90**	**63.46**	**53.29**	**53.87**
CRD+KD	71.97	70.74	68.83	66.84	61.27	59.07	48.58	47.94
CRD+L^2RKD	**72.82**	**71.51**	**69.43**	**68.91**	**63.85**	**62.58**	**52.95**	**53.82**
AB+KD	70.04	69.95	67.75	67.47	61.74	62.69	53.76	60.69
AB+L^2RKD	**73.36**	**72.78**	**70.80**	**71.47**	**65.58**	**68.70**	**59.41**	**64.07**
NST+KD	71.15	69.36	67.92	66.87	62.37	61.04	55.05	52.49
NST+L^2RKD	**72.27**	**71.84**	**69.66**	**69.68**	**65.33**	**65.01**	**58.83**	**58.22**

observed that $k \in [2, 5]$ works reasonably well while the performance decreases substantially when k is greater than a threshold. The reason may be that when k is large, the region of the linear combination becomes large as well. For example, the linear combination of three one-dimension points is a triangle region while the linear combination of of two one-dimension points is a linear region. Thus, the randomly sampled samples from a large region have a higher probability of being far from the real data distribution than those from a small region. The samples far from the real data distribution are less useful and even can be noise, which leads to substantial decreases in the performance.

Conclusion

Only distilling knowledge at sparse data points cannot enable a student to well capture the local shape of a teacher function. To address this issue, we have proposed L^2RKD which transfers the knowledge in local, linear regions from a teacher to a student. This is achieved by a regularizer which enforces the student to mimic the local shape of the teacher in linear regions. Compared with more than 10 state-of-the-art approaches, L^2RKD achieves a much better performance and is more data-efficient. Moreover, L^2RKD is compatible with these approaches to further improve their performances.

References

Aguilar, G.; Ling, Y.; Zhang, Y.; Yao, B.; Fan, X.; and Guo, E. 2019. Knowledge Distillation from Internal Representations. *arXiv preprint arXiv:1910.03723* .

Ahn, S.; Hu, S. X.; Damianou, A.; Lawrence, N. D.; and Dai, Z. 2019. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 9163–9171.

Andreas, J.; Rohrbach, M.; Darrell, T.; and Klein, D. 2016. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705* .

Anil, R.; Pereyra, G.; Passos, A.; Ormandi, R.; Dahl, G. E.; and Hinton, G. E. 2018. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235* .

Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. A. 2019. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 5049–5059.

Chapelle, O.; Weston, J.; Bottou, L.; and Vapnik, V. 2001. Vicinal risk minimization. In *Advances in neural information processing systems*, 416–422.

Chen, D.; Mei, J.-P.; Wang, C.; Feng, Y.; and Chen, C. 2020. Online Knowledge Distillation with Diverse Peers. In *Thirty-Forth AAAI Conference on Artificial Intelligence*.

Cho, J. H.; and Hariharan, B. 2019. On the efficacy of knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, 4794–4802.

Chung, I.; Park, S.; Kim, J.; and Kwak, N. 2020. Feature-map-level Online Adversarial Knowledge Distillation. In *Proceedings of International Conference on Machine Learning*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Guo, H.; Mao, Y.; and Zhang, R. 2019. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3714–3722.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D.; Mu, N.; Cubuk, E. D.; Zoph, B.; Gilmer, J.; and Lakshminarayanan, B. 2019. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781* .

Heo, B.; Kim, J.; Yun, S.; Park, H.; Kwak, N.; and Choi, J. Y. 2019a. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, 1921–1930.

Heo, B.; Lee, M.; Yun, S.; and Choi, J. Y. 2019b. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3779–3787.

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

Huang, Z.; and Wang, N. 2017. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219* .

Kim, J.; Park, S.; and Kwak, N. 2018. Paraphrasing complex network: Network compression via factor transfer. In *Advances in Neural Information Processing Systems*, 2760–2769.

Koratana, A.; Kang, D.; Bailis, P.; and Zaharia, M. 2019. LIT: Learned intermediate representation training for model compression. In *International Conference on Machine Learning*, 3509–3518.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Li, J.; Socher, R.; and Hoi, S. C. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394* .

Liu, Y.; Cao, J.; Li, B.; Yuan, C.; Hu, W.; Li, Y.; and Duan, Y. 2019. Knowledge Distillation via Instance Relationship Graph. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7096–7104.

Ma, N.; Zhang, X.; Zheng, H.-T.; and Sun, J. 2018. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 116–131.

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3967–3976.

Passalis, N.; and Tefas, A. 2018. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 268–284.

Peng, B.; Jin, X.; Liu, J.; Li, D.; Wu, Y.; Liu, Y.; Zhou, S.; and Zhang, Z. 2019. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, 5007–5016.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2015. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations*.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Serban, I. V.; García-Durán, A.; Gulcehre, C.; Ahn, S.; Chandar, S.; Courville, A.; and Bengio, Y. 2016. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus. *arXiv preprint arXiv:1603.06807* .

Simard, P. Y.; LeCun, Y. A.; Denker, J. S.; and Victorri, B. 1998. Transformation invariance in pattern recognition—tangent distance and tangent propagation. In *Neural networks: tricks of the trade*, 239–274. Springer.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

Srinivas, S.; and Fleuret, F. 2018. Knowledge Transfer with Jacobian Matching. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 4723–4731. Stockholmsmässan, Stockholm Sweden: PMLR.

Thulasidasan, S.; Chennupati, G.; Bilmes, J. A.; Bhattacharya, T.; and Michalak, S. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in Neural Information Processing Systems*, 13888–13899.

Tian, Y.; Krishnan, D.; and Isola, P. 2020. Contrastive representation distillation. In *International Conference on Learning Representations*.

Tung, F.; and Mori, G. 2019. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, 1365–1374.

Vapnik, V. N. 1998. *Statistical learning theory*. J. Wiley.

Wang, X.; Zhang, R.; Sun, Y.; and Qi, J. 2018. Kdgan: Knowledge distillation with generative adversarial networks. In *Advances in Neural Information Processing Systems*, 775–786.

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

Zagoruyko, S.; and Komodakis, N. 2016. Wide Residual Networks. In *BMVC*.

Zagoruyko, S.; and Komodakis, N. 2017. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018a. mixup: Beyond Empirical Risk Minimization. *International Conference on Learning Representations* URL https://openreview.net/forum?id=r1Ddp1-Rb.

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

Zhu, X.; Gong, S.; et al. 2018. Knowledge distillation by on-the-fly native ensemble. In *Advances in neural information processing systems*, 7517–7527.