Adaptive Teacher Finetune: Towards high-performance knowledge distillation through adaptive fine-tuning

Zhenyan Hou 1, a, Wenxuan Fan 2, b

1 International School of Information Science & Engineering (DUT-RUISE), Dalian University of Technology, Dalian, China
2 Dalian University of Technology, Dalian, China.

a houzhenyan@foxmail.com, bfanwenxuan1@foxmail.com

Abstract. Knowledge distillation is a widely used method to transfer knowledge from a large model to a small model. Traditional methods use pre-trained large models to supervise the training of small models, called Offline Knowledge Distillation. However, the structural gap between teachers and students limits its performance. After that, Online Knowledge Distillation retrained the teacher-student network from the beginning and the method of echo teaching greatly improved the performance. But there is very little work to explore the difference between the two. In this paper, we first point out that the essential difference between Offline and Online Knowledge Distillation is actually whether the weight of the teacher-student network has a process of mutual adaptation. If they adopt the teacher network and the student network jointly train to implement Offline Knowledge Distillation, there is no obvious difference in the final performance, no matter whether it is a joint distillation training. This shows that teacher-student network adaptation is important for Knowledge Distillation. Then, we propose an Adaptive Teacher Finetune (ATF) to adapt the teacher model to the student network. It will use student model information for Finetune during the Offline Knowledge Distillation process. With normalized logical distribution and alpha-divergence, the performance improvement of ATF clearly exceeds the existing Offline and Online Knowledge Distillation method. Extensive experiments conducted on cifar and ImageNet support our aforementioned analysis and conclusions. With the newly introduced ATF, we obtained state-of-the-art performance on ResNet 18 on ImageNet.

1. Introduction

Knowledge distillation [1] is a well-known technique to learn compact deep neural network models with competitive accuracy, where a smaller network (student) is trained to simulate the representations of a larger one (teacher) with higher accuracy. The popularity of knowledge distillation is mainly due to its simplicity and generality; it is straightforward to learn a student model based on a teacher and there is no restriction about the network architectures of both models. The main goal of most approaches is how to transfer dark knowledge to student models effectively, given predefined or pretrained teacher networks. Parameter quantification or character recognition, pruning, and knowledge distillation (KD) are some well-known methods in this subject [1]. KD defines the class probability of the teacher system as the aim that the tiny contactors strives to replicate by transferring the knowledge of big pre-training
networks (or integration of tiny networks) as a teacher network. Students can enhance their performance by matching their predictions with the teacher’s predictions. There is no unique teacher-student function provided to users refinement. Starting with the initial training, all systems learn at the same time by teaching each other. It uses ground truth labels to train traditional cross-entropy loss and imitation loss to learn from peers. The result obtained by the network trained in this online distillation method is not only better than the network trained only by cross-entropy loss, but it performs better than a network taught in the typical offline distillation method through an also before the teacher network Hinton et al. developed many prospective knowledge distillation approaches to promote the optimization process of distilling by using diverse "information," such as intermediate representation, in [2], inter-layer flow [3], pay attention to figure [5], structure Relationship [6] and activation similarity [7]. While these system well in compressed deployment models, they often use a two-stage training approach, which involves before the a highly capable previous teachers before transferring knowledge to a compact student model, which takes more time and money. Without pre-training Highly capable instructor, online data extraction [10,9,8,11] uses a phased end-to-end training strategy to improve the target network for deployment in the case of knowledge extraction across several networks or branches. To improve the generalization ability of a hierarchical network without adding additional reasoning cost, it is recommended to divide knowledge among many classifier heads, and is unable to create a more powerful online teacher to help pupils improve their grades. The work of Lan et al. is more similar to ours. [10] To expand the adaptability of each student network, use a low- and mid network and collect login information from various branches (students) with teachers as the unit. Regardless, logit accumulation obstructs additional integration teacher optimization, and online integration overlooks branch collaboration, resulting in poor performance. Kim et al. [9] combine various branch embedding into online collections, however their approaches require additional convolutional filters for feature fusion, and they are unable to take use of branch collaboration.

Although deep neural networks have shown extraordinary capabilities in various applications, their superior performance comes at the cost of high storage and computing costs. Therefore, the acceleration and compression of neural networks have caused widespread concern. Knowledge Transfer (KT) is a popular solution that aims to train a smaller network of students by transferring knowledge from a larger teacher model. This article gives the promotion of KL-divergence, introducing the rigor of the student network as a continuous parameter. So as to achieve this goal, we adopt the concept of alpha-divergence, by minimizing these distance metrics as a loss function, and generalizing the knowledge distillation algorithm based on student networks with different degrees of rigor. Compared with the corresponding method based on Shannon entropy, this method can significantly improve the performance of the student network. We verified the effectiveness of our method in multiple data sets, and further compared it with other KT methods to verify the superiority of the method.

As far as we know, KT’s earliest work can be traced back to. They trained a compression model that contains fake data labeled by a set of strong classifiers. However, their work is limited to shallow models. Until recently, Hinton et al. brought it back by introducing knowledge distillation (kd). The basic idea of Kd is to learn the category distribution provided by teachers through the softened Softmax, extract knowledge from the large teacher model, and form a pupil model. Despite its simplicity, kd has shown good results in various image classification tasks. In this work, we propose a more reasonable loss function for knowledge distillation and extend KL-divergence to alpha-divergence. Different authors use the alpha parameter in different ways. Figure 1 describes the alpha-divergence of Zhu & Rohwer (1995). In order to understand how the choice of alpha affects the result of knowledge distillation, consider the problem of minimizing alpha-divergence to approximate a complex distribution p by a traceable Gaussian distribution q. Figure 1 shows (non-standardized) approximations for different alpha values. This shows that when alpha is a large positive number, q tends to cover all modes of p, while for alpha tends to negative infinity (assuming divergence is finite), q is attracted to the mode with the highest probability quality. We tested our method on the cifar-10, cifar-100, and imagenet datasets, and the results showed that our Alpha-Divergence Knowledge Distillation significantly improved students’ academic performance. In summary, the contributions of this work are as follows: • We
introduce alpha-divergence for knowledge distillation as a promotion of KL-divergence. Then, we discussed the connection with KL-divergence, thus demonstrating the richness of this new loss function family. • We analyzed the loss function of students approaching the teacher network and gave theoretical guarantees.

![Figure 1](image.png)

*Figure 1. The Gaussian q which minimizes a -divergence to p (a mixture of two Gaussians), for varying a. α → -∞ prefers matching one mode, while α → ∞ prefers covering the entire distribution.*

Experiments show that the suggested ATF improves accuracy in a variety of neural networks and datasets in a consistent and meaningful way. Experiments using ten different types of neural networks on five different changing technology that ATF outperforms the state-of-the-art distillation approach on both image classification and image classification by a considerable margin. On average, accuracy boosts of 5.46 percent, 1.71 percent, 1.18 percent, 1.25 percent, and 0.82 percent were recorded on CIFAR100, CIFAR10, and ImageNet datasets. Aside from that, an ablation research and a hyper-parameters sensitivity study are carried out to demonstrate the efficacy and durability of AID.

2. Related work

In addition to probability distribution, other studies also try to extract various features for students [9], [10], [11], [12], [13], [14]. In terms of model efficiency of training tiny student networks, KD is also regarded one of several model compression approaches including such trimming and quantization [15], [16]. Deep mutual learning (DML) [17] proposes using Kullback Leibler divergence (KLD) to train students' networks to exchange messages, which outperforms the original network in terms of performance. In this paradigm, each student network acts as a teacher network for other student networks. This approach has the advantage of being able to adapt diverse network designs with ease. However, because it does not leverage the significant data in the instructor model to refine, this strategy can provide scant information for the aim. Dynamic local integration (one) [18] is an online distillation method that only trains an add multiple network and uses branch logic gating to develop a powerful instructor model to improve students' network learning. This strategy pulls knowledge from the instructor network and transfers it to the student network in one direction. It can only be used if the branches have the same architecture because it uses gating modules on the same layer. Furthermore, in many visual tasks, this method based on logit cannot make full use of the valuable particular operation. Knowledge extraction is also employed in our FFL to increase subnetwork performance and generate relevant feature maps.

B. In feature fusion methods study [19], [7], feature fusion is used in dual learning. Figure 2 shows two alternative outputs from bilinear CNN [19]: Fusion module structure M filters are used to execute deep convolution on the subnet's cascaded feature map. Then, for point-by-point convolution, n filters are needed. The network is fused to produce a bilinear vector. Dualnet [7] employs the "sum" method to integrate the features of two parallel networks with the very same architecture to construct a fusion classifier. It also employs iterative training to learn complimentary characteristics by updating the weights of sub networks alternatively. There are three differences between our FFL and dualnet. For starters, dualnet only works with the same subnet design, whereas FFL works with any network design. Second, we hope that the trainable fusion module outperforms the traditional feature fusion method (see Figure 2). Finally, dualnet only focuses on improving the performance of the fusion classifier, whereas FFL focuses on improving the performance of both the fusion classifier and the sub network through mutual knowledge extraction.
3. Online KD

Online knowledge extraction: although the traditional media method requires pre-training of a teacher model, the online method does not. Instead, these networks teach each other by sharing knowledge throughout the training process. DML is a contemporary example of an open approach that has shown promising outcomes. The DML simply applies KD loss to all teachers as a result of which it outperforms the offline KD technique. DML’s downside is that it lacks a true teaching role, limiting the amount of information it can supply to every network. This flaw in DML has been identified. It is preferable to build the gating set logistic regression of the training network and utilize it as the realignment goal of each network rather than extracting one another and between networks. An endeavor to develop a strong instructor logic can yield a more diverse set of data. For example, the gating component shares the underlying structure and is unable to train several network architectures at nearly the same time. The most serious flaw in present customized services is that they rely solely on logit and ignore feature location information. Given that the KD loss item is only valid to classification problems, transfer of knowledge in the ideations layer has the potential to broaden the applicability to other tasks. As a result, we offer a separation approach that not only employs logit and moreover employs specialized via adversarial training, and which can be used to a variety of collaborating training access technologies.

Domain adaptation

Domain adaptation is part of the transfer learning field. In its most popular setting, the goal of domain adaptation is to improve test performance on unlabeled target domains, while the model is built on related but different source domains. In the literature, two KL-divergence-based metrics, JS-divergence and alpha-divergence, have been successfully used to measure domain similarity in domain adaptation research, and they are calculated based on the data distribution of the domain. Existing definitions of alpha-divergence other than Zhu & Rohwer (1995) above include: Amari

\[
D_{\alpha}[p \parallel q] = \frac{1}{\alpha - 1} \left( 1 - \int p(\theta)^{\alpha-1} q(\theta)^{-\alpha} d\theta - 1 \right)
\]

\[
D_{\alpha}[\rho \parallel \theta] = \frac{1}{\alpha - 1} \log \int \rho(\theta)^{\alpha-1} \theta^{-\alpha} d\theta
\]

4. Method

4.1. Overview

The conventional knowledge distillation approaches attempt to find the way of teaching student networks given the architecture of teacher networks. Since the teacher network is trained only with the loss with respect to the groundtruth, and the optimization of the objective is not necessarily beneficial.
for knowledge distillation to students. To the contrary, SFTN framework aims to improve the effectiveness of knowledge distillation from the teacher to the student models. The procedure of SFTN is composed of the following steps.

4.2. Revisit of Offline and Oneline Knowledge Distillation Approaches

Distillation Neural folding network usually generates class edge probability by playing "softmax" along the output layer, which converts the logit $*$ and probability of each class by playing along% I and comparing with other folds.

$$ q_j = \frac{\exp(z_j/T)}{\sum_i \exp(z_i/T)} $$ (1)

Where $t$ is the temperature normally set to 1. Using a higher $t$ value produces a softer probability distribution across categories.

In the simplest distillation form, knowledge is transferred to distillation model by training on transfer set and using soft target distribution in each case. The same high temperature is used in training distillation models, but 1 C after training. When all or part of the right label of the transfer combination is known, this method can be significantly improved by training distillation models to produce the correct label. One way is to modify software targets using the correct tags, but it is better to find that only two weighted averages of different target functions are used. The first objective function is the cross entropy with soft target, and the cross entropy is calculated in softmax of distillation model with the same high temperature, which is used to generate some soft targets from the model. The second objective function is the cross entropy with the correct label.

$$ L_{C1} = -\sum_{i=1}^{N} \sum_{m=1}^{M} I(y_i, m) \log(p_i^m(x_i)) $$ (2)

With an indicator function $I$ defined as

$$ I(y_i, m) = \begin{cases} 1 & y_i = m \\ 0 & y_i \neq m \end{cases} $$ (3)

Conventional supervised loss training networks can correctly predict the labels of training instances. To enhance the generalisation efficiency on the done depending, we use another peer network e2 to provide training experience in the form of its posterior probability $p2$. To quantify the match of the two network's predictions $p_i$ and $p_0$, we use the Kullback Leibler (KL) Divergence. The KL distance from $p_i$ to $p_2$ is computed as

$$ D_{KL}(p_2 || p_1) = \sum_{i=1}^{N} \sum_{m=1}^{M} p_2^m(x_i) \log \frac{p_2^m(x_i)}{p_1^m(x_i)} $$ (4)

The overall loss functions $L_e$ and $L_{e2}$ for networks e and e2 respectively are thus

$$ L_{e} = L_{C1} + D_{KL}(p_2 || p_1) $$ (5)

$$ L_{e2} = L_{C2} + D_{KL}(p_1 || p_2) $$ (6)

4.3. Our Adaptive Teacher Finetune Approach

Our methods for offline KD and online KD are as follows:

For offline KD

The first step is to train the teacher network: $T$: $\text{loss}_T = \text{loss	extunderscore ce}(T)$

The second step is to train the student network: $S$: $\text{loss}_S = \text{loss	extunderscore ce}(S) + \text{loss	extunderscore kl(\text{pretrain}(T)\rightarrow S)}$

For online KD 2. Train the teacher-student model in one step
For the teacher network: \( T: \text{loss}_t = \text{loss}_{ce}(T) + \text{loss}_{kl}(S\rightarrow T) \)
For the student network: \( S: \text{loss}_s = \text{loss}_{ce}(S) + \text{loss}_{kl}(T\rightarrow S) \)

Table 1. The main difference in this one is pretain (T) and loss_kl (T->S) We do experimental analysis separately

| Method               | pretain(T) | loss_kl (S->T) | Top-1  |
|----------------------|------------|----------------|--------|
| offline KD           | Y          | NO             | 70.52  |
| online KD            | NO         | Y              | 71.62  |
| online KD wo(S->T)   | NO         | NO             | 70.41  |
| Ours ATF             | Y          | Y              | 71.89  |

It can be seen that loss_kl (S->T) is very important for KD, so we use loss_kl (S->T) for Finetune on the basis of offline KD

For offline KD
The first step is to train the teacher network: \( T: \text{loss}_t = \text{loss}_{ce}(T) \)
The second step is to train the student network: \( S: \text{loss}_s = \text{loss}_{ce}(S) + \text{loss}_{kl}(\text{pretain}(T)\rightarrow S) \)

5. Ours ATF uses loss_kl (S->T) for Finetune for pretain(T)
The second step is to train the student network:
For the teacher network: \( T: \text{loss}_t = \text{loss}_{ce}(T) + \text{loss}_{alpha}(S\rightarrow T \text{ pretain}(T)) \)
For student network: \( S: \text{loss}_s = \text{loss}_{ce}(S) + \text{loss}_{alpha}(T\rightarrow S) \)

5.1. Normalized logistic distribution and alpha-divergence
KL divergence has been widely used to measure the discrepancy in output probabilities between the teacher and student models in KD. One main drawback with KL divergence is that it cannot sufficiently penalize the student model when it over-estimates the uncertainty of the teacher model. Let \( p \) and \( q \) denote the output probability of the teacher and student models, respectively. The KL divergence between the teacher and student models is calculated by \( KL(p||q) = Ep[\log p/q] \). When \( p > 0 \), to ensure \( KL(p||q) \) remains finite, we must have \( q > 0 \). This is the zero avoiding property of KL. In contrast, when \( p = 0, q > 0 \) does not get penalized. For example, as shown in Figure 2 (b) and (c), even though the student model overestimates the uncertainty of the teacher model and predicts the wrong class ("class 4"), the KL divergence is still small. The aforementioned over-estimation in Example 2 would be penalized at a larger magnitude when using other types of divergences, e.g., reverse KL divergence \( KL(q||p) \). For reverse KL divergence, \( KL(q||p) = Eq[\log q/p] \) is infinite if \( p = 0 \) and \( q > 0 \). Hence if \( p = 0 \) we must ensure \( q = 0 \), this is known as the zero forcing property (Murphy, 2012). Therefore, minimizing reverse
KL divergence encourages the student model \( q \) to avoid low probability modes of \( p \) while focusing on the modes with high probabilities, and thus, may under-estimate the uncertainty of the teacher model, as shown in Example 1 in Figure 2. Hence, a natural question is whether it is possible to generalize the KL divergence to simultaneously suppress both the under-estimation and over-estimation of the teacher model uncertainty during the supernet training.

### 3.2. KD with adaptive \( \alpha \)-divergence

Our observations shown in Figure 2 motivate us to design a new KD objective that simultaneously penalize both overestimation and under-estimation of the teacher model uncertainty. We first generalize the typical KL divergence with a more flexible \( \alpha \)-divergence (Minka et al., 2005). Consider \( \alpha \in \mathbb{R} \setminus \{0, 1\} \), the \( \alpha \)-divergence is defined as

\[
D_{\alpha}(p \parallel q) = \frac{1}{\alpha(\alpha - 1)} \sum_{i=1}^{m} q_i \left[ \left( \frac{p_i}{q_i} \right)^\alpha - 1 \right]
\]

Where \( q = [q_i]_{i=1}^{m} \) and \( p = [p_i]_{i=1}^{m} \) are two discrete distributions on \( m \) categories. The \( \alpha \)-divergence includes a large spectrum of classic divergence measures. In particular, the KL divergence \( KL(p \parallel q) \) is the limit of \( D_{\alpha}(p \parallel q) \) with \( \alpha \rightarrow 1 \) while the reverse KL divergence \( KL(q \parallel p) \) is the limit of \( D_{\alpha}(p \parallel q) \) with \( \alpha \rightarrow 0 \). A key feature of \( \alpha \)-divergence is that we can decide to focus on penalizing different types of discrepancies (underestimation or over-estimation) by choosing different \( \alpha \) values. For example, as shown in Figure 2 (c), when \( \alpha \) is negative, \( D_{\alpha}(p \parallel q) \) is large when \( q \) is more widely spread than \( p \) (when \( q \) over-estimates the uncertainty in \( p \)), and is small when \( q \) is more concentrated than \( p \) (when \( q \) underestimates the uncertainty in \( p \)). The trend is opposite when \( \alpha \) is positive: under-estimation would be more heavily penalized than over-estimation.

To simultaneously alleviate the over-estimation and underestimation problem when training the supernet, we consider a positive \( \alpha^+ \) and a negative \( \alpha^- \), and propose to use the maximum of \( D_{\alpha^+}(p \parallel q) \) and \( D_{\alpha^-}(p \parallel q) \) in the KD loss function:

\[
D_{\alpha^+,\alpha^-}(p \parallel q) = \max\{D_{\alpha^+}(p \parallel q), D_{\alpha^-}(p \parallel q)\}
\]

penalizing over-estimation penalizing under-estimation

### 5.2. Experiment Setting

Classification of images. Nine different types of deep neural networks are used in the picture classification tests. Each model in the CIFAR experiment is trained with 300 epochs by the SGD optimizer, with a batch size of 128. Each model is trained with 90 epochs by the SGD optimizer in ImageNet experiments, with a batch size of 256.

Experiments using comparisons. For comparison, four different types of knowledge distillation methodologies were used. All of these experiments were carried out by us.

**Table 3.** Experiment results on CIFAR

| Model          | Baseline | KD  | FitNet | DML  | SD   | TOFD |
|----------------|----------|-----|--------|------|------|------|
| ResNet18       | 94.25    | 94.67 | 95.57  | 95.19 | 95.87 | 96.92 |
| ResNet50       | 94.69    | 94.56 | 95.83  | 95.73 | 96.01 | 96.84 |
| PreactResNet18 | 94.20    | 93.74 | 95.22  | 94.80 | 95.08 | 96.49 |
| PreactResNet50 | 94.39    | 93.53 | 94.98  | 95.87 | 95.82 | 96.93 |
| SEResNet18     | 94.78    | 94.53 | 95.64  | 95.37 | 95.51 | 96.80 |
| SEResNet50     | 94.83    | 94.80 | 95.31  | 94.83 | 95.45 | 97.02 |
| ResNeXt50-4    | 94.49    | 95.41 | 95.78  | 95.41 | 96.01 | 97.09 |
| MobileNetV1    | 90.16    | 91.70 | 90.53  | 91.65 | 91.98 | 93.93 |
| MobileNetV2    | 90.43    | 92.86 | 90.49  | 90.49 | 91.02 | 93.34 |
| ShuffleNetV1   | 91.33    | 92.57 | 92.23  | 91.40 | 92.47 | 92.73 |
| ShuffleNetV2   | 90.88    | 92.42 | 91.83  | 91.87 | 92.51 | 93.74 |
Table 4. Experiment results on CIFAR10 (Top-1 Accuracy %). Numbers in bold are the highest.

| Model         | Baseline | TOFD    | MAC(G) | Param(M) |
|---------------|----------|---------|--------|----------|
| ResNet18      | 69.76    | 70.92   | 1.82   | 11.69    |
| ResNet50      | 76.13    | 77.52   | 4.11   | 25.56    |
| ResNet101     | 77.37    | 78.64   | 7.83   | 44.55    |
| ResNet152     | 78.31    | 79.21   | 11.56  | 60.19    |
| ResNeXt50-32-4| 77.62    | 78.93   | 4.26   | 25.03    |
| WideResNet50-2| 78.47    | 79.52   | 11.43  | 68.88    |

5.3. Results on CIFAR10 and CIFAR100

The efficiency of student networks on CIFAR100 and CIFAR10 is shown in Tables 2 and 3. It is discovered that: (a) When compared to baseline models, the proposed TOFD provides a significant accuracy boost. In CIFAR100, the eleven models had an average accuracy boost of 5.46 percent, ranging from 6.75 percent at SENet50 to 3.78 percent at ShuffleNetV1 as the minimum. In CIFAR10, the eleven models had an average accuracy boost of 2.49 percent, ranging from 3.77 percent at MobileNetV1 to 1.40 percent at ShuffleNetV1 as the lowest. (b) The suggested TOFD exceeds the second best distillation method by a significant margin in all models. On CIFAR100 and CIFAR10, respectively, accuracy increases by 3.13 percent and 1.28 percent when compared to the second best distillation method. (c) The suggested TOFD is effective in lightweight models like MobileNet and ShuffleNet as well as over-parameters models like ResNet and SENet. On the CIFAR100 and CIFAR10 datasets, the lightweight models provide an average accuracy improvement of 4.40 percent and 2.74 percent, respectively.

5.4. Results on ImageNet

Table 4 illustrates the outcomes of TOFD's ImageNet experiment. In all of these studies, the ResNet152 model is used as the instructor model. The results show that (a) TOFD improves accuracy by 1.18 percent on average across the 6 neural networks. (b) The ResNet50 and ResNet101 distilled versions are more accurate than the ResNet101 and ResNet152 baselines, respectively. TOFD provides 1.57 times compression and 1.81 speed with no loss of accuracy by substituting purified ResNet50 and ResNet101 with ResNet101 and ResNet152, respectively.

References

[1] Hinton, G., Vinyals, O., and Dean, J. Distilling the Knowledge in a Neural Network. In NeurIPS Deep Learning and Representation Learning Workshop, 2015.
[2] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks, 2017. 1, 3, 4
[3] Feng Wang, Xiang Xiang, Jian Cheng, and Alan Loddon Yuille. Normface. Proceedings of the 2017 ACM on Multimedia Conference - MM ’17, 2017.
[4] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 3962–3971, 2019. 6
[5] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 7130–7138, 2017. 6
[6] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 7089–7097, 2019. 2
[7] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. ArXiv, abs/1910.10699, 2020. 1, 6
[8] Sungsoo Ahn, Shell Xu Hu, Andreas C. Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. 2019 IEEE/CVF Conference on
Computer Vision and Pattern Recognition (CVPR), pages 9155–9163, 2019. 6

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. 2

[10] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[11] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 3962–3971, 2019. 6

[12] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. ArXiv, abs/1707.01219, 2017. 6

[13] Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 1345–1354, 2019. 1, 2

[14] Jangho Kim, SeongUk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer, 2020. 6

[15] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples, 2018. 1

[16] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6738–6746, 2017. 2