# Joint Regularization Knowledge Distillation

Haifeng Qing[1], Ning Jiang[1,2(✉)], Jialiang Tang[3], Xinlei Huang[1,2], and Wengqing Wu[1]

[1] School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621000, Sichuan, China
jiangning@swust.edu.cn
[2] Jiangxi Qiushi Academy for Advanced Studies, Nanchang 330036, Jiangxi, China
[3] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, Jiangsu, China

**Abstract.** Knowledge distillation is devoted to increasing the similarity between a small student network and an advanced teacher network in order to improve the performance of the student network. However, these methods focus on teacher and student networks that receive supervision from each other independently and do not consider the network as a whole. In this paper, we propose a new knowledge distillation framework called Joint Regularization Knowledge Distillation (JRKD), which aims to reduce network differences through joint training. Specifically, we train teacher and student networks through joint regularization loss to maximize consistency between the two networks. Meanwhile, we develop a confidence-based continuous scheduler method (CBCS), which divides examples into center examples and edge examples based on the example confidence distribution of network output. Prediction differences between networks are reduced when training with a central example. Teacher and student networks will become more similar as a result of joint training. Extensive experimental results on benchmark datasets such as CIFAR-10, CIFAR-100, and Tiny-ImagNet show that JRKD outperforms many advanced distillation methods.

**Keywords:** Knowledge Distillation · Joint regularization · Continuous scheduler

## 1 Introduction

Over the past few decades, deep neural networks (DNNs) have enjoyed great success in computer vision fields [20,25], such as real-time semantic segmentation [7], object detection [15]. However, powerful DNNs frequently have larger parameters and require large computational and storage resources, which are undesirable for industrial applications. To address this issue, a number of model compression techniques have been proposed, including model pruning [2,16,30], quantification [9], and knowledge distillation [5], with knowledge distillation proving to be a mature method for improving the performance of small models.


B. Luo et al. (Eds.): ICONIP 2023, CCIS 1966, pp. 173–184, 2024.
https://doi.org/10.1007/978-981-99-8148-9_14

Traditional knowledge distillation (KD) [5] (see Fig. 1(a)) utilizes a soft label from a pretrained teacher to supervise students to obtain similar performance to the teacher, which is a two-stage training process and not flexible. Recently, online knowledge distillation [14,25] proposed a single-stage scheme to encourage networks to train each other and retrain teacher and student networks to improve consistency on different points of view [1,4,23]. For example (see Fig. 1(b)), deep mutual learning (DML) [22] predictions after the classifier of teacher and student. Chung *et al.* [3] introduces the middle layer feature transition between teacher and student, as shown in Fig. 1(c). The existing online knowledge distillation method is a way of teaching and learning collaboratively, and we hope to further enhance this collaboration, bringing students and teachers together as a whole.

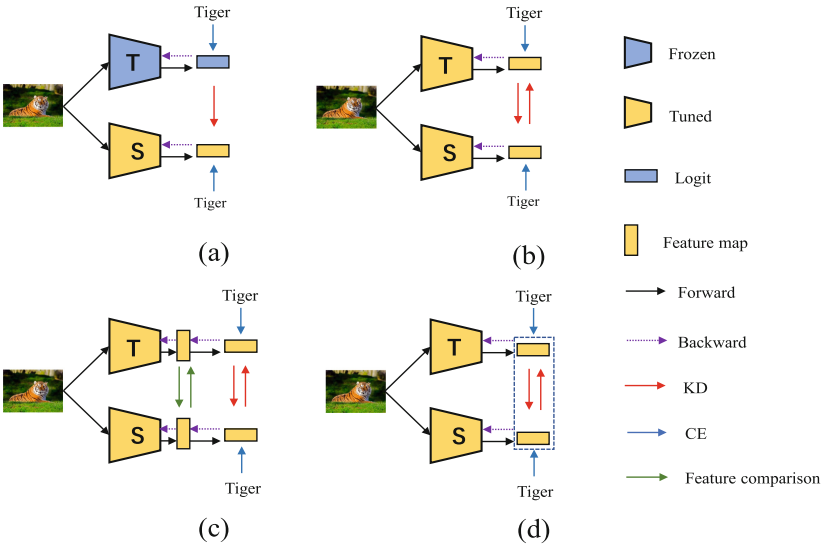


**Fig. 1.** Illustration of (a) KD, (b) DML, (c) DML with Feature comparison, and (d) Knowledge distillation framework with joint regularization loss.

When the differences between teacher and student models are too great, distillation can adversely affect students [17]. Strengthening connections between teacher and student networks can improve distillation performance, from traditional knowledge distillation to online knowledge distillation (see Sect. 4.2,"Proof"). Recently, Wei *et al.* [29] proposes a robust federated learning method called Jocor to maximize similarity between DNNs by reducing their. Based on this insight, we believe that the teacher and student can obtain consistent joint supervision in predictions, enhancing the integrity of the two classifiers, and thus improving the distillation performance, as shown in Fig. 1(d).

This paper proposes a new knowledge distillation framework called Joint Regularization Knowledge Distillation (JRKD). Specifically, we train teacher

and student networks through joint losses to maximize consistency between the two networks. Inspired by "course learning" [21], we propose a confidence-based continuous scheduler method (CBCS), which divides examples into center examples and edge examples based on their-confidence density distributions calculated teacher and student. The central example reduces the prediction error in the joint training of the two networks, promotes their mutual learning, and reduces the accumulation of error streams in the network. The proportion of central examples gradually increases as the training process progresses, ensuring the integrity of the training set. Extensive experiments on three representative benchmarks have shown that our JRKD can effectively train a high-performance student network.

1. We propose a joint regularized knowledge distillation method(JRKD), which can effectively reduce the differences between networks.
2. We used federated regularized loss to normalize teacher and student networks to maximize consistency across networks.
3. We develop a confidence-based continuous scheduling method (CBCS), through which the selection of loss instances can mitigate the negative impact between networks and reduce the difficulty of consistency training.

## 2 Related Literature

In this section, we will discuss the work related to online knowledge distillation and Disagreement. In both areas, various approaches have been proposed over the past few years. We summarize it below.

### 2.1 Online Knowledge Distillation

Traditional knowledge distillation is achieved by a network of pre-trained teachers who take their knowledge (extracted logits [5] or intermediate feature forms [11]) and guide students to models during the training process. This method is simple and effective, but it requires a high-performance teacher model. In online knowledge distillation, the teacher and student models update their own parameters at the same time to achieve an end-to-end process. The concept of online distillation was first proposed by Zhang et al. [22] to enable collaborative learning and mutual teaching between students and teachers. In order to address the impact between network mutual learning, SwitKD [25]adaptively calibrates the gap during the training phase through a switching strategy between the two modes of expert mode (pause teacher, keep student learning) and learning mode (restart teacher), so that teachers and students have appropriate distillation gaps when learning from each other. Chung et al. [26] adds a feature-map-based judgment to the original logit-based prediction, and the feature-map-based loss controls the teacher and student to distill each other through the adjudicator.

Online distillation is a single-stage training scheme with efficient parallel computation. The existing online knowledge distillation method is a way of teaching and learning collaboratively, and we hope to further enhance this collaboration, bringing students and teachers together as a whole, rather than individually.

### 2.2 Disagreement

Weakly supervised learning [27] solves the problem of time-consuming and labor-intensive collection of large and accurate data sets, and the use of online queries and other methods will inevitably be affected by noise labels. In recent years, the "divergence" strategy has been introduced to address such issues. For example, decoupling [19]uses two different networks, and when there is no difference in the predictions of the two networks, the network parameters are not updated, and the network is updated when there is a disagreement. "Divergence" strategy expectations use these examples that produce different predictions to steer the network away from current errors. In 2019, Chen et al [12] combined the "divergence" strategy with Co- teaching [28] in collaborative teaching to provide good performance in terms of DNN's robustness to noise tags. Recently, Wei et al. [29] proposed a robust learning paradigm called JoCoR from different perspectives, which aims to reduce the diversity of training examples of two networks during training, and update the parameters of two networks at the same time by selecting examples with small losses. Under the training of joint loss, the two networks will become more and more similar due to the effects of coregularization.

We hope to be able to use the idea of "divergence" strategy in the field of knowledge distillation, aiming to reduce the differences between teacher and student networks, thereby improving the integrity between networks and improving distillation performance.

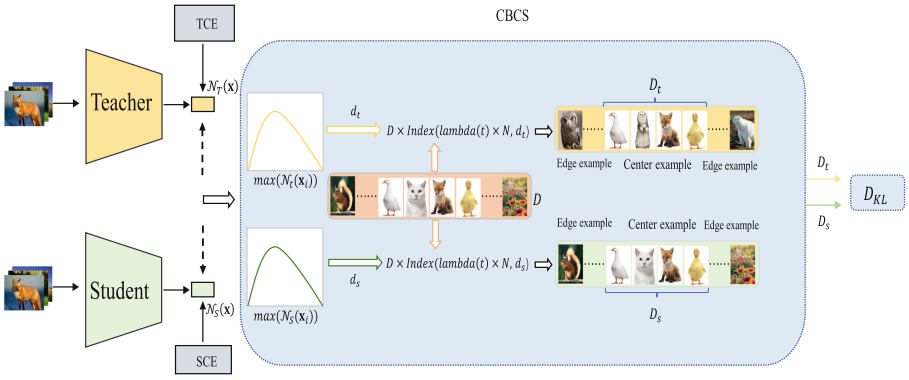


**Fig. 2.** JRKD flowchart, CBCS selects a central example based on the network output's example confidence, and teachers and students receive joint supervision training.

## 3 Approach

In this section, we will discuss how CBCS selects central examples (Sect. 3.1) and how joint regularization loss trains the network collaboratively (Sect. 3.2).

### 3.1 Confidence-Based Continuous Scheduler

According to recent research [29], while networks can improve consistency between them through joint regularization, they are vulnerable to error streams caused by biased selection. To address this issue, we design a confidence-based continuous scheduler (CBCS) that divides the example into center examples and edge examples. Using central example training can better reduce the prediction bias between networks. This is shown in Fig. 2.

Different center examples are chosen by teachers and students; we only show how teachers choose, and students do the same. We use dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{n}$ as the network input for each batch with $n$ examples. Let the teacher network be $\mathcal{N}_T$, and the prediction probability on the dataset $\mathcal{D}$ be $\mathcal{N}_T(\mathbf{x}_i)_{i=1}^{n}$. For Class $m$ classification tasks, $\max(\mathcal{N}_S(\mathbf{x}_i))$ represents the maximum confidence that the teacher network prediction instance $\boldsymbol{x}_i$ is for one of the classes in class $m$. The KMeans clustering algorithm is used to obtain the maximum confidence centroid of $n$ examples:

$$M\_p_{\text{target}} = \frac{\sum_{i=1}^{N} \max(\mathcal{N}_S(\mathbf{x}_i))}{N}. \tag{1}$$

$M\_p_{\text{target}}$ is the centroid of confidence. Calculating the absolute distance from each $\max(\mathcal{N}_S(\mathbf{x}_i))$ to $M\_p_{\text{target}}$ yields the set $d_t = [d_1, d_2, \ldots, d_n]$. The smaller the value in $d_t$, the closer to the confidence center.

CBCS controls the central example proportion for each period through a continuous scheduling functions $lambda(t)$. $T_{\text{total}}$ is the total training cycle, $\lambda_0$ Represents the proportion of the initial central example selection, $t$ stands for epock currently trained:

$$lambda(t) = \min\left(1, \lambda_0 + \frac{1 - \lambda_0}{T_{\text{total}}} \cdot t\right). \tag{2}$$

By using $d_t$ as the basis for selecting the central example, the example size is controlled by the $lambda(t)$. Index$(\cdot, \cdot)$ is a function method that returns an index of multiple minimum values. Get the current set of central example $D_t$, the same process can also obtain $D_s$:

$$D_t = D \times Index\left(\text{lambda}(t) \times N, d_t\right), \quad D_t \in D, \tag{3}$$

$$D_s = D \times Index\left(\text{lambda}(t) \times N, d_s\right), \quad D_s \in D. \tag{4}$$

### 3.2 Joint Regularization Knowledge Distillation

For the multi-class classification task for class $m$. We use two deep neural networks to express the proposed JRKD method. For clarity, we set $p_s = \left[p_s^1, p_s^2 \ldots, p_s^m\right]$ and $p_t = \left[p_t^1, p_t^2, \ldots, p_t^m\right]$ as the final prediction probabilities of the example $\boldsymbol{x}_i$ by students and teachers, respectively. It is obtained by softening the network output by the $softmax$ function of distillation temperature $T = 3$.

**Joint Regularization Loss.** We train the two networks together using joint regularization loss, which brings the predictions of each network closer to the peer-to-peer network. Under joint training, networks will become more and more similar to each other. To accomplish this, asymmetric Kullback-Leibler (KL) divergence is used:

$$\mathcal{L}_{con} = D_{\mathrm{KL}}\left(\boldsymbol{p}_s \| \boldsymbol{p}_t\right) + D_{\mathrm{KL}}\left(\boldsymbol{p}_s \| \boldsymbol{p}_t\right). \tag{5}$$

$\mathcal{L}_{con}$ represents the joint regularization loss. CBCS selects different central examples to participate in joint training based on the confidence probability of the examples generated by teachers and students:

$$D_{\mathrm{KL}}\left(\boldsymbol{p}_s \| \boldsymbol{p}_t\right) = \sum_{i=1}^{N} \sum_{m=1}^{M} p_s^m\left(\boldsymbol{x}_i\right) \log \frac{p_s^m\left(\boldsymbol{x}_i\right)}{p_t^m\left(\boldsymbol{x}_i\right)}, x \in D_t, \tag{6}$$

$$D_{\mathrm{KL}}\left(\boldsymbol{p}_t \| \boldsymbol{p}_s\right) = \sum_{i=1}^{N} \sum_{m=1}^{M} p_t^m\left(\boldsymbol{x}_i\right) \log \frac{p_t^m\left(\boldsymbol{x}_i\right)}{p_s^m\left(\boldsymbol{x}_i\right)}, x \in D_s. \tag{7}$$

**Total Losses.** For JRKD, the joint regularization loss is used to improve the integrity between the networks, and the conventional supervision loss is used to maintain the correctness of the learning. JRKD minimizes the following losses to train the network:

$$\mathcal{L}_T = \mathcal{L}_{TCE} + \mathcal{L}_{con}, \tag{8}$$

$$\mathcal{L}_S = \mathcal{L}_{SCE} + \mathcal{L}_{con}. \tag{9}$$

$\mathcal{L}_{SCE}$ and $\mathcal{L}_{TCE}$ represent conventional supervision loss for students and teachers, respectively. Finally, we give the algorithm flow table of JRKD, as shown in Algorithm 1.

**Algorithm 1.** JRKD

**Input:** Network $f$ with $\boldsymbol{\Theta} = \{\boldsymbol{\Theta}_t, \boldsymbol{\Theta}_s\}$, learning rate $\eta$, fixed $\tau$, epoch $T_k$ and $T_{\max}$, iteration $I_{\max}$;

**for** $t = 1, 2, \ldots, T_{\max}$ **do**
  Shuffle training set $D$;
  **for** for $n = 1, \ldots, I_{\max}$ **do**
    Fetch mini-batch $D_n$ from $D$ ;
    $p_s = f_s\left(\boldsymbol{x}, \boldsymbol{\Theta}_s\right), \forall \boldsymbol{x} \in D_n$ ;
    $p_t = f_t\left(\boldsymbol{x}, \boldsymbol{\Theta}_t\right), \forall \boldsymbol{x} \in D_n$ ;
    Calculate the example size by (2) from $lambda(t)$;
    Obtain training subset $D_s$,$D_t$by (3,4)from $D_n$ ;
    Obtain $L_S$,$L_T$ by $(8, 9)$ on $D_s, D_t$ ;
    Update $\boldsymbol{\Theta t} = \boldsymbol{\Theta t} - \eta \nabla L_T, \boldsymbol{\Theta s} = \boldsymbol{\Theta s} - \eta \nabla L_S$;
  **end for**
**end for**

**Output:** $\boldsymbol{\Theta}_s$ and $\boldsymbol{\Theta}_t$

## 4 Experiments

In this section, we select three representative image classification tasks for experiments in Sect. 3.1 to evaluate the performance of JRKD. The ablation experiment at Sect. 3.2 confirmed the effectiveness of CBCS and loss of joint regularization. In addition, we analyze the effect of $\lambda_0$ initial center example ratio on performance. In Sect. 3.3, visualize the probability distribution of teacher and student network outputs.

**Experiment Setup.** The configuration of our experiment is to descend SGD with a stochastic gradient and set the learning rate, weight decay, and momentum to 0.1, $5 \times 10^{-4}$, and 0.9, respectively. The dataset uses a standard data augmentation scheme and normalizes [17] the input image using channel means and standard deviations.

### 4.1 Experiments on Benchmarks

**Results on Tiny-ImageNet.** It contains 200 categories, each containing 500 training images, 50 validation images, and 50 test images. After using JRKD, the two groups of networks obtained an accuracy of 59.43% and 55.71%, respectively. It can effectively improve the accuracy of the student network. Compare these methods, Our method also achieves good results. The results are shown in Table 1.

**Table 1.** The accuracy of the comparison method comes from the papers of other authors. JRKD verified accuracy results on the Tiny-ImageNet dataset.

| Teacher | ResNet34 | WRN40-2 |
|---|---|---|
| Student | MobileNetV2 | ResNet20 |
| DML | 55.70 | 53.98 |
| KDCL | 57.79 | 53.74 |
| SwitOKD [25] | 58.79 | 55.03 |
| JRKD | **59.43** | **55.71** |

**Results on CIFAR-100.** The CIFAR-100 dataset has 100 classes. Each class has 500 sheets as a training set and 100 as a test set. Table 2 shows the experimental results, and JRKD outperforms many other methods on various network architectures. Impressively, JRKD achieves 1.33% (WRN-40-2/WRN-16-2) accuracy improvement to DML on CIFAR-100. Besides, JRKD also shows 0.88% and 0.19% (ResNet32 $\times$ 4/ResNet8 $\times$ 4) accuracy gain over ReviewKD and DKD, respectively.

**Results on CIFAR-10.** The CIFAR-10 dataset has a total of 60,000 examples, which are divided into 50,000 training examples and 10,000 test examples. The

**Table 2.** JRKD verified accuracy results on the CIFAR-100 dataset. W40-2, R32x4, R8x4 and SV1 stand for WRN-40-2, ResNet32 × 4, ResNet8 × 4,ShuffleNetV1. The accuracy of other methods is mainly derived from DKD [22].

| Teacher<br>Student | W40-2<br>W16-2 | W40-2<br>SV1 | R32 ×4<br>R8 ×4 | VGG13<br>VGG8 |
|---|---|---|---|---|
| Teacher | 75.61 | 75.61 | 79.42 | 74.64 |
| Student | 73.26 | 70.50 | 72.50 | 70.36 |
| KD [5] | 74.92 | 74.83 | 73.33 | 72.98 |
| FitNets [8] | 73.58 | 73.73 | 73.50 | 71.02 |
| RKD [10] | 73.59 | 72.21 | 71.90 | 71.48 |
| CRD [6] | 75.48 | 76.05 | 75.51 | 73.94 |
| AT [11] | 74.08 | 73.32 | 73.44 | 71.43 |
| CC [13] | 75.66 | 71.38 | 72.97 | 70.71 |
| DML [18] | 75.33 | 75.58 | 74.30 | 73.64 |
| KDCL [14] | 74.25 | 74.79 | 74.03 | 71.26 |
| ReviewKD [19] | 76.12 | 77.14 | 75.63 | N/A |
| DKD [22] | 76.24 | 76.70 | 76.32 | 74.68 |
| JRKD | **76.66** | **77.24** | **76.51** | **74.90** |

experimental results are shown in Table 3, using the same experimental configuration as other methods. Our method not only improved student performance, the teacher achieved an accuracy gain of 0.23% and 0.8% over SwithOKD and KDCL, respectively.

**Table 3.** Ours results are the average over 5 trials. Comparison of performances with powerful distillation techniques using the same 200 training epochs. Performance metrics refer to the original article.

| | Backbone | KDCL | SwithOKD | JRKD |
|---|---|---|---|---|
| Student | WRN-16-1 | 91.86 | 92.50 | **93.11** |
| Teacher | WRN-16-8 | 95.33 | 94.76 | **95.56** |

### 4.2 Ablation Experiments

CIFAR-100 was chosen for the dataset of the ablation experiment. As shown in Table 4, we quantified the gap between teachers and networks using T-S gap, and compared KD and DML, JRKD can effectively reduce the differences between networks and improve distillation performance. The JRKD† compared other distillation methods and showed that joint regularization loss can improve similarity between networks. The comparison of JRKD and JRKD† shows that CBCS

is beneficial for online training. In addition, the sensitivity analysis of the $\lambda_0$ parameter manually set in the continuous scheduler $lambda(t)$ was performed. As shown in Table 5, The value of $\lambda_0$ in the continuous scheduler generally defaults to 0.3, so we only analyze the value around 0.3 and find that the appropriate $\lambda_0$ is conducive to distillation.

**Table 4.** Verify the effectiveness of joint regularization losses and CBCS. The student network is MobileNetV2, the teacher is the VGG13, $KD_{T\to S}$ represents the teacher network to accept student supervision, Top-1 is the classification accuracy of CIFAR-100, T-S gap uses KL to calculate the gap between output logical values between networks. JRKD† refers to the absence of CBCS to select loss instances.

| Method | $KD_{T\to S}$ | $KD_{S\to T}$ | Top-1 | T-S gap |
|---|---|---|---|---|
| KD | ✔ | ✘ | 67.37 | 1.12 |
| DML | ✔ | ✔ | 68.52 | 0.83 |
| JRKD† | ✔ | ✔ | 69.12 | 0.61 |
| JRKD | ✔ | ✔ | 69.55 | 0.42 |

**Table 5.** The parameter sensitivity experiment of the continuous scheduler of CBCS. The experimental data set uses CIFAR-100, and the experimental accuracy result is averaged 5 times.

| | $\lambda_0$ | 0.2 | 0.3 | 0.4 |
|---|---|---|---|---|
| Student | WRN-16-2 | 76.35 | **76.66** | 76.43 |
| Teacher | WRN-40-2 | 78.40 | **78.89** | 78.47 |

### 4.3 Visual Analytics

We compare the traditional online knowledge distillation method DML and the JRKD by feeding the same batch of examples into the trained network and visualizing the confidence distribution of the examples by the teacher-student network. As shown in Fig. 3, the confidence distribution of the teacher-student network is more similar in the example output of JRKD, demonstrating that JRKD can improve network similarity.

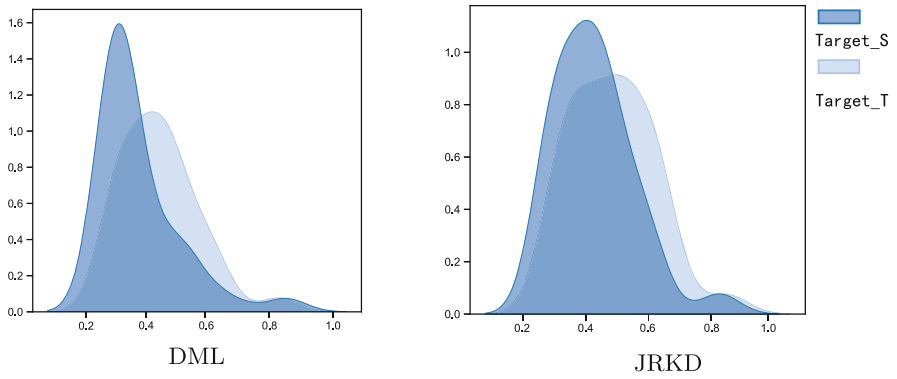


**Fig. 3.** Two different methods produce confidence profiles.

## 5 Conclusion

This paper proposes an effective method called JRKD to reduce the differences between networks. The key idea of JRKD is to train the teacher and student networks by jointly regularizing losses to maximize consistency between the two networks. In order to reduce the difficulty of federation, we developed a confidence-based continuous scheduling method (CBCS), which can divide samples into central samples and edge samples according to the sample confidence distribution of network output. In the early stage of joint training, when training with central examples, the prediction difference between networks is reduced, and edge samples are added to the training with the training cycle to ensure the integrity of the training samples. We demonstrated the effectiveness of JEKD with a large number of experiments, and analyzed the joint regularization loss and the training aid of CBCS through ablation experiments. In future work, we will continue to explore the correlation between teacher networks and student networks as a whole training in online knowledge distillation.

**Acknowledgement.** This research is supported by Sichuan Science and Technology Program (No. 2022YFG0324), SWUST Doctoral Research Foundation under Grant 19zx7102.

## References

1. Smith, J., et al.: Always be dreaming: a new approach for data-free class-incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
2. Jiang, Y., et al.: Model pruning enables efficient federated learning on edge devices. IEEE Trans. Neural Netw. Learn. Syst. (2022)
3. Chung, I., Park, S., Kim, J., Kwak, N.: Feature-map-level online adversarial knowledge distillation. In: ICML (2020)

4. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 92–100 (1998)
5. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
6. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. In: ICLR (2020)
7. Wang, Y., et al.: LEDNet: a lightweight encoder-decoder network for real-time semantic segmentation. In: 2019 IEEE International Conference on Image Processing (ICIP). IEEE (2019)
8. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: FitNets: hints for thin deep nets. In: ICLR (2015)
9. Faisant, N., Siepmann, J., Benoit, J.-P.: PLGA-based microparticles: elucidation of mechanisms and a new, simple mathematical model quantifying drug release. Eur. J. Pharm. Sci. **15**(4), 355–366 (2002)
10. Park, W., Lu, Y., Cho, M., Kim, D.: Relational knowledge distillation. In: CVPR (2019)
11. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In: ICLR (2017)
12. Yu, X., Han, B., Yao, J., Niu, G., Tsang, I.W., Sugiyama, M.: How does disagreement benefit co-teaching? arXiv preprint arXiv:1901.04215 (2019)
13. Peng, B., et al.: Correlation congruence for knowledge distillation. In: ICCV (2019)
14. Guo, Q., et al.: Online knowledge distillation via collaborative learning. In: CVPR (2020)
15. Choi, H., Bajić, I.V.: Latent-space scalability for multi-task collaborative intelligence. In: 2021 IEEE International Conference on Image Processing (ICIP). IEEE (2021)
16. Li, B., Wu, B., Su, J., Wang, G.: EagleEye: fast sub-net evaluation for efficient neural network pruning. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12347, pp. 639–654. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58536-5_38
17. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. PMLR (2015)
18. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: CVPR (2018)
19. Malach, E., Shalev-Shwartz, S.: Decoupling "when to update" from "how to update". In: Advances in Neural Information Processing Systems, pp. 960–970 (2017)
20. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (2012)
21. Wang, X., Chen, Y., Zhu, W.: A survey on curriculum learning. IEEE Trans. Pattern Anal. Mach. Intell. **44**(9), 4555–4576 (2021)
22. Zhao, B., et al.: Decoupled knowledge distillation. In: CVPR (2022)
23. Tanaka, D., Ikami, D., Yamasaki, T., Aizawa, K.: Joint optimization framework for learning with noisy labels. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5552–5560 (2018)

24. Sindhwani, V., Niyogi, P., Belkin, M.: A co-regularization approach to semi-supervised learning with multiple views. In: Proceedings of ICML Workshop on Learning With Multiple Views, pp. 74–79 (2005)
25. Qian, B., Wang, Y., Yin, H., Hong, R., Wang, M.: Switchable online knowledge distillation. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13671, pp. 449–466. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20083-0_27
26. Chung, I., et al.: Feature-map-level online adversarial knowledge distillation. In: International Conference on Machine Learning. PMLR (2020)
27. Zhou, Z.-H.: A brief introduction to weakly supervised learning. Natl. Sci. Rev. **5**(1), 44–53 (2018)
28. Han, B., et al.: Co-teaching: robust training of deep neural networks with extremely noisy labels. In: Advances in Neural Information Processing Systems, vol. 31 (2018)
29. Wei, H., et al.: Combating noisy labels by agreement: a joint training method with co-regularization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
30. Tang, J., et al.: Data-free network pruning for model compression. In: 2021 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE (2021)