


DCD: A New Framework for Distillation Learning With Dynamic Curriculum

1st Jiachen Li 2nd Yuchao Zhang  * 3rd Yiping Li 4th Xiangyang Gong 5th Wendong Wang
BUPT BUPT University of Washington BUPT BUPT
Beijing, China Beijing, China Washington, USA Beijing, China Beijing, China
jiachen.li@bupt.edu.cn yczhang@bupt.edu.cn markli@uw.edu xygong@bupt.edu.cn wdwang@bupt.edu.cn

Abstract—Deep neural network(DNN) has exhibited outstanding performance in many tasks like computer vision(CV) and natural language processing(NLP). The pursuit of even higher performance pushes neural networks to grow wider and deeper with numerous parameters. A notable drawback of such large models with billions of parameters is the ever-increasing demands on computing and memory, which makes these models unable to be deployed on resource-constrained systems. Knowledge distillation(KD) is a representative technique to reduce the number of parameters by transferring knowledge from a complex model to a lightweight model.

In this paper, we doubt the widely used random sampling process ignores the difference among all the heterogeneous samples damages distillation performance, and argue that samples should be presented in a specific order. But how to feed samples to the distilling process remains an open question. To address the above problem, we present DCD - an automatic distillation learning framework where training samples are arranged by a dynamic curriculum strategy. Such a specific sequential training method imitates the human learning process and thus achieves faster convergence speed and higher accuracy.

We evaluated the performance of DCD using three image classification datasets CIFAR-10, CIFAR-100, and CINIC-10. The results show that DCD improves the network accuracy by 1.7%, 2.5%, and 1% respectively. Moreover, we also showed the same effectiveness of DCD in the case of noise, which makes it more practical in reality and universal in other scenarios.

Index Terms—knowledge distillation, curriculum learning , difficulty indicator, training schedule, model compression

I. INTRODUCTION

In recent years, deep neural network algorithms have shown strong power in various areas such as computer vision (CV) [1] and natural language processing (NLP) [2]. However, to meet the requirements of high accuracy applications, neural network structures are becoming more and more complex with an ever-increasing amount of parameters. Such large models are impractical to be deployed to edge devices.

To solve the above problem, researchers have proposed many attempts to compress those large models to reduce the computational overhead and storage space. As a representative

model compression strategy, knowledge distillation (KD) [3] has achieved remarkable performance improvements and therefore received rapidly increasing attention in recent years. The main idea of knowledge distillation is to train a lightweight student networks to mimic the knowledge extracted from the original network to reduce model size. KD does perform well in model compression, but does not in final performance. So here we put forward a question, how to train lightweight student models to achieve higher accuracy?

Inspired by human education system, which is a customized orderly earning progress, students learn from easy tasks to challenging ones. We doubt the random sampling process in traditional KD methods and argue that samples should be presented to the distillation framework in a specific order. This idea coincides with curriculum learning (CL) [4] which optimize the performance of traditional machine learning (ML) algorithm in a similar way. The curriculum learning strategy has improved the accuracy of NLP task by 1.37% on TNEWS [5], and CV task by 1.81% on ImageNet [6].

However, the curriculum strategy that works on traditional ML cannot be used on KD directly. Designing a sample strategy for KD still facing two key technical challenges. The first challenge is to precisely measure the difficulty of tasks. Measure sample difficulty in KD has an intrinsic trade-off between the teacher network indicator and student network indicator because the student network learns from both the teacher network and instances. The student network is insufficient to measure sample difficulty at the beginning of KD process, and the curriculum arranged by a large teacher network may not be suitable for the student network. The second challenge is to generate sample sequences. It is challenging to predefine a universal schedule function because the performance of the student network is affected by task characteristics, distillation framework, and network structure. On the other hand, introducing heuristic or meta-learning algorithms as schedulers will cause mass extra computing costs. To address the above challenges, we proposed a distillation learning framework with dynamic curriculum called DCD, which consists of a curriculum module and a distillation module, the distillation process alternates between the curriculum module and distillation module until the student network converges. The curriculum module is responsible for generating a sample

*Corresponding Author: Yuchao Zhang

The work was supported in part by the National Natural Science Foundation of China(NSFC) under Grant 62172054, the National Key R&D Program of China under Grant 2019YFB1802603, the Key Project of Beijing Natural Science Foundation under M21030, the NSFC under Grant 62072047, and the BUPT-ChuangCache Joint Laboratory Project under Grant A2022164.

sequence based on a fixed distillation snapshot. In detail, we first employ the loss of teacher network simulation and the confidence of the student network on samples as difficulty indicators and weight them according to student network performance. Then we introduce a scheduler base on student network performance feedback to determine the sample subset for the next distillation step. Regarding the distillation module, the student network first trains on the sample subset generated by the curriculum module, then updates the snapshot of the student network, which is used in the curriculum module as an indicator. We highlight that DCD is a general plug-in application that can apply to different knowledge distillation methods.

We conducted comprehensive experiments to validate the performance of our approach and compared DCD with various curriculum strategies on CIFAR-10, CIFAR-100, and CINIC-10. The result shows that DCD improved accuracy by 1.7%, 2.5%, and 1%, respectively. We then evaluate DCD in noisy scenarios, and the experimental results prove that DCD improves student network accuracy by 1.71%, 1.68%, and 1.58% with acquisition noise.

Our contributions can be summarized as follows:

- We disclose the necessity of introducing curriculum learning in knowledge distillation rather than feeding training samples randomly throughout the distillation process.
- We proposed a data-driven curriculum module to automatically feed samples into the distillation process in a specific order by employing the feedback of the student network and the snapshot of distillation networks.
- We conducted a series of experiments to demonstrate the optimization of our approach and fully discussed the applicable scenarios and the choice of key hyper-parameters in experiments.

The rest of the paper is structured as follows: related works are presented in Section II, then the background and the motivation are given in Section III, in Section IV we introduce the framework and each module in detail, and the experimental evaluation is provided in Section V, conclusions are included in Section VI.

II. RELATED WORK

Our work connects two recently emerged research hotspots of machine learning. First, knowledge distillation is a representative neural network compression technique. The vanilla distillation framework claims teacher network output contains more information than original labels [3], which can be used to train the student network because additional abstracted knowledge is more suitable for neural network learning [7]. Therefore, FITNET [2] employs the outputs of intermediate layers to train the student network. AT [8] introduce the attention map as the soft label to provide supervised information. To further enhance KD performance, FSP [9] allows the student network to mimic the features matrix calculated as the inner product between feature maps from teacher network layers. Besides, feature embedding has been

used in student network training [10]. Instead of extracting knowledge from a single teacher network, multiple teacher frameworks are also proposed to improve the performance of the student network [11], [12]. In terms of sample strategy on KD, researchers proposed Data-Free distillation, which transfers knowledge to the student network on the generated samples to overcome the reliance on sample collection [13], [14]. Knowledge distillation have been widely used in many fields, including image classification tasks like interpretation and diagnosis tasks [15], visual recognition [16] and text-to-image synthesis [17]. natural language processing (NLP) tasks like natural language understanding(NLU) [18], text generation [19] and text recognition [20]. recommendation tasks like [21] and speech recognition tasks to improve the efficiency and recognition accuracy of acoustic models [22], [23] and many other applications [24]–[26].

Secondly, curriculum learning(CL) gradually attracted the attention of researchers. Inspired by the human education system, CL improves model performance by feeding samples in a meaningful order rather than random. Original CL [27] proposed a predefined complexity indicator and presented low complex examples first and gradually introduced more complex samples in training process. Rather than the predefined indicator, self-pace-learning [28] takes the sample-wise loss of the target model snapshot as indicator to reduce the dependence on prior knowledge and achieve automatic CL. Furthermore, self-paced-curriculum-learning [29] proposed a general framework and exploited the combination of prior knowledge and learner feedback as training scheduler. To further improve the flexibility of the curriculum framework, the teacher-student framework allows student network learning on the instance sequence scheduled by the independent teacher model [30]. From the optimization perspective, researchers combined CL with reinforcement learning(RL) to adjust the curriculum through a trainable meta learner network [31]. Recent years, CL is widely used in various application scopes, including supervised learning tasks within computer vision [32], natural language processing [33], healthcare prediction [34], and reinforcement learning [35].

III. BACKGROUND AND MOTIVATION

We first discuss the background of KD and CL in this Section. Next, we disclose that distillation on ordered samples can effectively improve student network performance through a case study.

A. Knowledge Distillation

Knowledge Distillation introduces the soft label predicted by a heavy but top-performance teacher model to provide extra information to train a lightweight student network. The student network mimics the output vector of the teacher network, which encodes the similarities between different categories as

$$p^i(x; \tau) = \frac{e^{s_i(x)/\tau}}{\sum_k e^{s_k(x)/\tau}} \quad (1)$$

where i is the category of sample x and $s_i(x)$ is the output distribution of sample x belongs to category i . In particular, a hyper-parameter τ is introduced to soften the output distribution of the teacher network and adjust the relative magnitude of the soft label. Distillation loss L_{KD} is invited to transfer knowledge by providing an additional loss signal as

$$L_{KD} = -\tau^2 \sum_{x \sim \mathcal{D}_x} \sum_{i=1}^C p^i(x; \tau) \log(p_s^i(x; \tau)) \quad (2)$$

where $p_s^i(x; \tau)$ and $p_t^i(x; \tau)$ represent the output of the student and the teacher. C is the total the number of categories and \mathcal{D}_x indicates the training set. The total loss L_{Total} of student network in KD is formulated as a linear combination of the standard cross-entropy loss L_{CE} between the student network output and one hot label and knowledge distillation loss L_{KD} as

$$L_{Total} = \lambda_1 L_{CE} + \lambda_2 L_{KD} \quad (3)$$

parameters λ_1 and λ_2 are introduced to balance two supervised signals.

B. Curriculum learning

Curriculum learning (CL) is a training strategy inspired by the learning process of human curricula by training a machine learning model in a meaningful order instead of randomly selected samples. The CL method has proven effective as a simple plug-in strategy to improve the generalization capacity and performance of model in various applications.

A conventional curriculum is a sequence of sample criteria throughout the training process: $C = S_1 \dots S_t \dots S_T$, curriculum criterion is also a sample re-weighting function, which typically meets three definitions. Firstly, the complexity of the training data should gradually increase in the training process, the re-weighting of examples in later steps increases the probability of more complex samples. Secondly, the amount of samples gradually increases during the training process. In the end, sample re-weights gradually coincide with the uniform distribution of the original dataset and degenerate to training on randomly selected samples.

C. Toy Experiment

We devised a simple experiment to demonstrate the value of distillation in a specific order. We simply rank samples $\{x_1, x_2, x_3 \dots x_{3n}\}$ according the classification loss l of the top-performance teacher network where $\{l_{x1} < l_{x2} < l_{x3}\}$. Then, we divide samples into three datasets where $x_1, x_2 \dots x_n \in Dataset_1$, $\{x_{n+1}, x_{n+2} \dots x_{2n}\} \in Dataset_2$, and $\{x_{2n+1}, x_{2n+2} \dots x_{3n}\} \in Dataset_3$. Correspondingly, the KD process is also divided into three stages. The student network training 10,000 steps on *Dataset1* at first, then we add *Dataset2* into the training set and perform the next 20,000 steps. In the end, *Dataset3* is added, and the student network learns on the original dataset until convergence. The baseline strategy is to distill on randomly selected samples, that is, randomly select sample batch on the dataset until convergence.

As shown in Figure1, distilling from easy to hard significantly improves distillation performance compared to the distillation on a randomly selected sample.

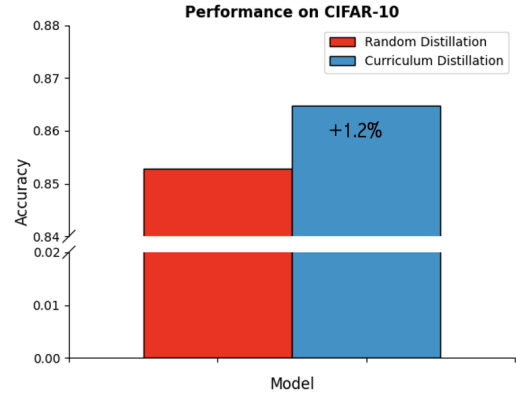


Fig. 1. Top-1 accuracy of random sample selection-based distillation versus curriculum-based distillation on CIFAR-10

IV. METHODOLOGY

In the following, we introduce the DCD framework and describe our proposed curriculum module in detail.

A. The Framework of Dynamic Curriculum Distillation

DCD is a novelty distillation framework for optimizing the effectiveness of KD with the help of curriculum strategies. Figure2 shows the distilling process of DCD. By applying DCD, curriculum and distillation modules run alternately until student network convergence on the training set. The curriculum module weighted sample based on the current state of the student network, which consists of two key components. 1) **Difficulty Indicator** measures the complexity of samples according to the student and teacher network snapshot. 2) **Training Scheduler** feed sample subsets to distillation in the specific order by updating weights vector based on fresh sample difficulty measurement. The distillation module trains the student network on the sample subset fed by the Curriculum module and updates the difficulty indicator correspondingly.

Formally, DCD framework aims to optimize the student model's parameter ω_s with pre-trained teacher network ϕ_t on dataset re-weighted by curriculum weight vector $v = [v_1, v_2 \dots v_n]$. The objective function is:

$$\min E_{\omega_s, v, \lambda, t} = \sum_{i=1}^n v_i l_i + R(v; \lambda) \quad (4)$$

Where l_i and v_i denote the total loss and curriculum weight in DCD of sample i . The notation R is a negative 11-norm curriculum regularization term formulated as

$$R(v; \lambda) = -\lambda \sum_{i=1}^n v_i \quad (5)$$

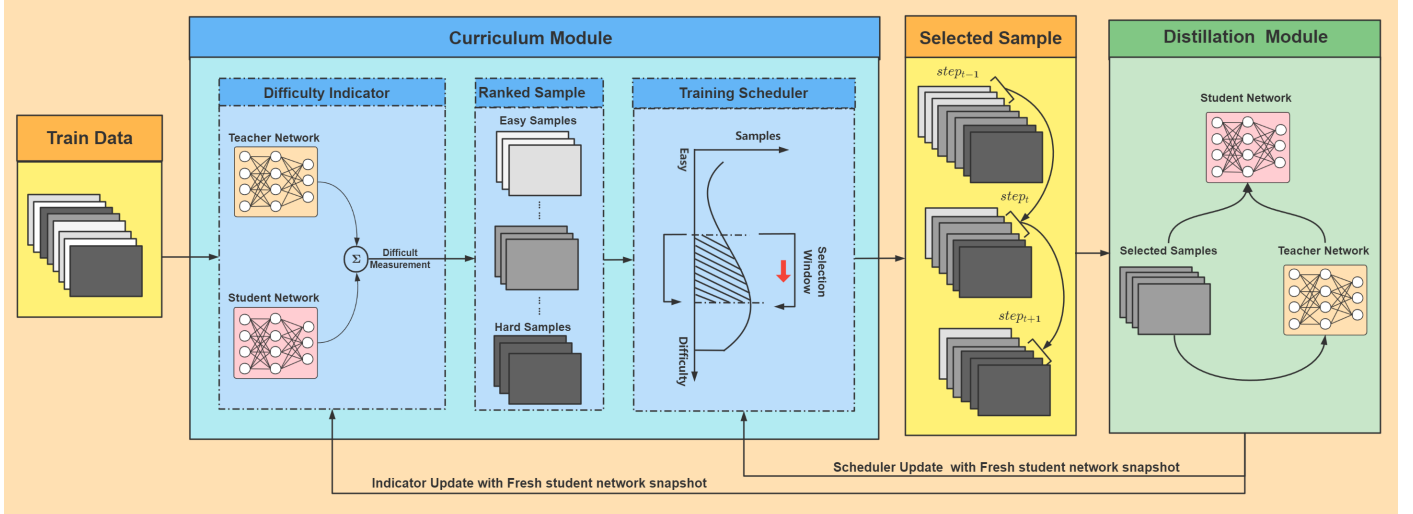


Fig. 2. The high-level view of the Dynamic Curriculum Distillation framework. DCD mainly consists of a curriculum module that selects samples based on the fresh snapshot of the teacher and student network and a distillation module that trains the student network on the selected samples.

In each distilling step, DCD first updates the weights vector with the snapshot of the student network and teacher network by solving:

$$v_i^* = \arg \min_{v_i \in [0,1]} v_i l_i + R(v; \lambda) \quad (6)$$

Next, the distillation module optimizes the student network parameter ω_s by solving the weighted distillation loss function:

$$w_s^* = \arg \min_{w_s} \sum_{i=1}^N v_i^* l_i \quad (7)$$

The overall algorithm of DCD is summarized in Algorithm 1.

B. Dynamic Difficulty Indicator

KD allows the student network learns directly from the sample while simulating the output of the teacher network. To reflect sample complexity accurately, we weighted two components of distillation loss formalized as

$$\phi_i = \omega_{CE} * L_{CE}(p_s^i, y_i) + \omega_{KD} * L_{KD}(p_s^i, p_t^i) \quad (8)$$

Where L_{CE} measures the complexity of a sample learned by the student network, and L_{KD} denotes the difficulty of the student network simulating the teacher network output on a sample. This function also can be used in the distillation step to update student network parameters ω_s to synchronize difficulty indicator and distillation loss in terms of weights assignment to both indicators. Next, We introduce the student network performance on the validation set Val_s to weight the first indicator as:

$$\omega_{CE} = \text{Max}(0, Val_s - 1/k) \quad (9)$$

Particularly, the number of categories k is invited to avoid overestimating student network performance because the initial

performance of the student network is affected by label distribution. Meanwhile, we weighted the second indicator I_D with the performance gap between teacher and student networks as

$$\omega_{KD} = \text{Max}(0, Val_t - Val_s) \quad (10)$$

Intuitively, the indicator changes during the distillation process. Simulation loss determines complexity indicator in the early distillation stage because the randomly initialized student network is unable to measure sample difficulty. When the student network is able to compete with the teacher, it is entirely up to the student network to measure the sample complexity.

C. Positive Training Scheduler

Next, we developed a novel performance-driven scheduler to automatically select samples for the next round. The main idea of our scheduler is adjusting sample difficulty according to the performance improved by training on current samples. If the performance improvement is considerable, distilling should continue on the current sample subset. Otherwise, the sample subset should be updated if the student model is already convergence on current samples. We introduced C to measure the marginal performance improvement by training the student network on the previous curriculum as

$$C^j = \text{Max}(Val_s^j - Val_s^{j-1}, 0) \quad (11)$$

We propose a positive training scheduler based on marginal performance with two characteristics. Firstly, we invite the order of the sample based on the complexity measurement rather than the absolute value in the scheduler because the difficulty measurement changes throughout the KD process. Next, when the student network converges on the current curriculum, the scheduler first updates the curriculum with the latest student network snapshot and previous complexity range.

Algorithm 1: Precoding of Dynamic Curriculum for Distillation

Input: Train set D_t , Val set D_v , Teacher network' parameter ω_t
Output: student network parameter ω_s^*
Initialize: ω_s , hyper-parameters s , distillation step j
 $Val_t \leftarrow$ Evaluate ω_t on validation data D_v
while $E(w_s, w_t, v)$ not converges **do**
 $Val_s \leftarrow$ validate ω_s^j on D_v
 $c_j \leftarrow C(Val_s^j, Val_s^{j-1})$
 if $s > c_j$ **then**
 $\phi_j \leftarrow$ Difficulty Indicator($\omega_s, \omega_t, Val_t, D_t$)
 $v_j \leftarrow$ Training Scheduler($\phi_j, c_j, D_t, \lambda_{j-1}$)
 end
 if $s > c_{j-1}$ and $s > c_j$ **then**
 $v_j \leftarrow$ Training Scheduler($\phi_j, c_j, D_t, \lambda_j$)
 end
 $l_j \leftarrow \phi_j$
 update $\omega_s^j = \arg \min_{\omega_s} \mathbb{E}(\omega_s, \omega_t, v_j, l_j)$ **return** ω_s^*
end

Function Training Scheduler():
 update λ_j by Eq.9-Eq.10
 for i in D_t **do**
 update v_i with λ_j and ϕ_i by Eq.11
 end
 $v_j \leftarrow [v_1^j, v_2^j, v_3^j \dots]$
 return v_j ;
End Function

Function Difficulty Indicator():
 $\alpha_S \leftarrow \text{Max}(0, Val_s^j - 1/k)$ by Eq.6
 $\alpha_D \leftarrow \text{Max}(0, Val_t - Val_s)$ by Eq.7
 for i in D_t **do**
 $I_S^i \leftarrow I_S(P_s^i, y_i)$
 $I_D^i \leftarrow I_D(P_s^i, P_t^i)$
 $\phi_i \leftarrow \alpha_S * I_S^i + \alpha_D * I_D^i$
 end
 $\phi_j \leftarrow [\phi_1^j, \phi_2^j, \phi_3^j \dots]$
 return ϕ_j ;
End Function

If there is still no performance improvement in the continuing distilling step, the sample weight function λ is updated as

$$\lambda_{upper}^j(C_j, \mu^j, \partial^j) = s \times \frac{\alpha}{C_v^j} \times (\mu^{j-1} + \partial^{j-1}) \quad (12)$$

$$\lambda_{lower}^j(C_j, \mu^j, \partial^j) = s \times \frac{\alpha}{C_v^j} \times (\mu^{j-1} - \partial^{j-1}) \quad (13)$$

Parameter μ^j denotes the lower bound of the sliding window at distilling step j , and ∂^j indicates the number of samples that should be taken into the sliding window, which is updated by the same amplitude simultaneously. We introduce a hyper-parameter s to control curriculum adjustment speed. Greater

s inclines a faster curriculum shift when sample contribution C is low and vice versa. According to fixed sample selection window, sample weight vector v is given by

$$v_i = \begin{cases} 1 & \lambda_{lower} < \phi_i < \lambda_{upper} \\ 0 & \text{else} \end{cases} \quad (14)$$

Intuitively, the sample subset difficulty adjusts automatically in the distillation process. If the student network improves performance by distilling on the current subset, the sample subset's difficulty increases slowly for thoroughly learning. Conversely, the difficulty of the sample subset is raised if the student network converges on it.

D. Distillation On Curriculum

Instead of distilling with random sample selection, DCD trains the student network with fixed sample weights v_j by solving the objective function as

$$w_s = \arg \min_{w_s} \sum_{i=1}^N v_i^j \text{Loss}_{DCD}(x_i) \quad (15)$$

where Loss_{DCD} is the distillation loss consisting of the distillation loss and the standard cross-entropy loss as

$$\text{Loss}_{DCD} = CE(P_s, y) + KL(P_s, P_t) \quad (16)$$

where the P_s and P_t refer to the outputs distribution of the student and teacher network. Standard cross-entropy loss is introduced to measure the mismatch of the ground truth label y and student network output. Kullback-Leibler divergence is used to measure the simulation loss between the student and teacher network.

V. EXPERIMENTS

In this section, we evaluated the performance of DCD and compared it with the predefined curriculum and random sample selection on three datasets, CIFAR-10, CIFAR-100, and CINIC-10. To make our experiments more comprehensive, we report the performance of DCD on data with varying levels of acquisition noise and further compare four learning rate function, which severely affects DCD performance. The performance of all the methods is evaluated by the Top-1 accuracy, which is the probability that the highest prediction probability category matches the actual category. All the experiments are implemented in TensorFlow 2.0 running on an AWS server equipped with an Intel Xeon E5-2630@2.6GHz and Tesla-T4 GPU.

A. Datasets and Backbone Network

In this paper, we employ three public image classification datasets, including CIFAR-10, CIFAR-100, and CINIC-10, to fully validate the performance of our approach. CIFAR-10 and CIFAR-100 [36] contain 50000 training images and 10000 image performance verification at resolution 32×32 RGB. CINIC-10 [36] is a more complicated classification task consisting of 270000 images from CIFAR and down-sampled ImageNet at resolution 32×32 RGB.

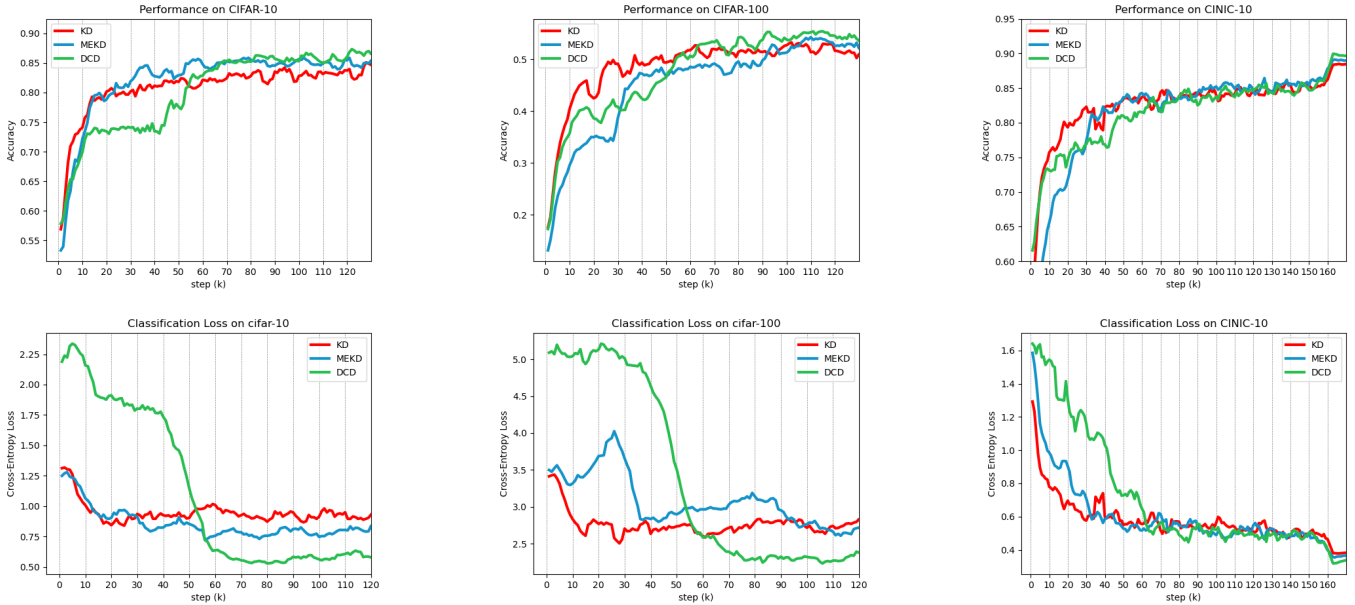


Fig. 3. The training loss and validation accuracy of KD, MEKD, and DCD on CIFAR-10, CIFAR-100, and CINIC-10 dataset

In terms of the KD framework, we adopt a response-based KD framework [3] as the backbone framework, which allows the student network simulates the output of the teacher network, the loss function of the student network is

$$l_i = (1 - \beta)CE(P_s^i, y_i) + \beta KL(P_s^i, P_t^i) \quad (17)$$

Where cross-entropy is used to measure the mismatch between the student network output and label, and KL-divergence is employed to measure the mismatch between student and teacher network output. Besides, the distillation temperature in all experiments is set to 3 for fairness. The backbone structure of the teacher network is ResNet-110 which has 1.7M parameters, a homogeneous but smaller structure-ResNet-20 is used for the student network with 0.27M parameters, which is only 15% that of the teacher network.

B. Compared Methods

We compare DCD with the following sample strategies commonly adopted in KD tasks.

- **Random curriculum** strategy is widely used in the conventional KD framework. Each distillation step randomly selects samples from the original training set to train the student network, so the distribution of samples used in each distillation step is consistent with that of the original training samples.
- **Predefined Multi-Evaluator Curriculum (MEKD)** employs teacher network simulation loss and student network confidence as complexity indicators with equal weight. By applying MEKD, the distilling process is divided into three stages. The student network distills on the simplest 1/3 sample in the first stage. Next, the student network snapshot in the complexity indicator is refreshed

to re-rank the sample and pick the 2/3 lower complexity samples for distilling till convergence. In the third stage, all samples are added to train the student network.

C. Main Result

We first compared the performance of DCD with the random curriculum KD and MEKD. In DCD, the hyper-parameter s is set to 0.01, and the distillation step is set to 1 epoch. In particular, we notice that the sample complexity also differs among categories. The category distribution of samples constantly changes in the distillation process, or even missing some category will damage the distillation performance seriously. Hence, we evaluate and schedule samples belonging to each category separately to ensure that the category distribution of samples in the training process is always consistent with the original distribution.

As shown in Table I DCD consistently outperforms the compared methods in accuracy on all three datasets. Details of the convergence curve and accuracy curve are shown in Figure 3 Firstly, Experiments show that DCD is an effective method for KD that improves accuracy by 1.7%, 2.5%, and 1% on CIFAR-10, CIFAR-100, and CINIC-10 compared to the conventional KD with random sample selection. Secondly, introducing MEKD into distillation can also improve the accuracy by 1.2%, 1%, and 0.2% on three datasets. These results also emphasize the importance of distilling the student network on samples in a specific order. Compared with MEKD, DCD further improves student accuracy by 0.5%, 1.5%, and 0.75%, mainly for two reasons. First, the dynamic weight allocation can balance two indicators better than the fixed one. Second, the performance-driven training scheduler can automatically adjust sample difficulty according to the student network

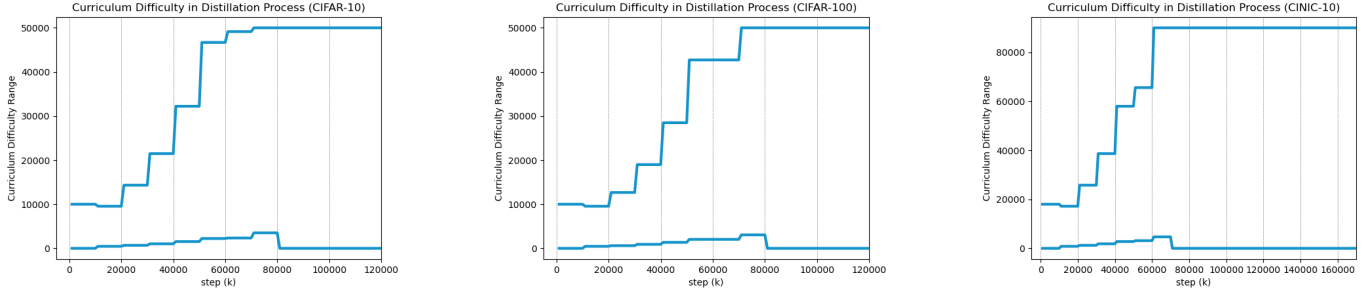


Fig. 4. The ranges of the sample difficulty throughout the DCD process on CIFAR-10, CIFAR-100, and CINIC-10.

feedback. The ranges of the sample difficulty throughout the DCD process are shown in figure 4.

TABLE I
VALIDATION ACCURACY COMPARISON BETWEEN PROPOSED METHOD AND BASELINES ON CIFAR-10, CIFAR-100, AND CINIC-10.

	Teacher	<i>KD</i>	<i>MEKD</i>	<i>DCD</i>
CIFAR-10	93.69%	85.29%	86.47%	87.00%
CIFAR-100	70.18%	55.08%	56.08%	57.58%
CINIC-10	92.71%	89.16%	89.37%	90.11%

D. Performance on data with acquisition noise

Data in reality applications is usually accompanied by acquisition noise, particularly prominent in applications like IoT services. Therefore, We further discuss whether the DCD method can improve performance in the presence of acquisition noise, which is particularly important for its application. We conduct experiments on the CINIC-10 dataset and simulate acquisition noise by adding random Gaussian noise to raw pictures. The hyperparameter α controls the intensity of the noise, which increases sequentially from 10 to 30 noise. In particular, we believe that the cost of obtaining a noise-free dataset with the same distribution as the original dataset is unacceptable in practice. Therefore the teacher network is also pre-train on samples with noise. As shown in the tableII, DCD improves the student network accuracy by 1.71%, 1.68%, and 1.63% on data with ever-increasing acquisition noise. Compared with distilling with random sample selection, DCD guides the optimization process more smoothly and more robust to mislabeled samples that helps the student network convergence to the global optimal.

TABLE II
PERFORMANCE OF DCD ON DATA WITH DIFFERENT DEGREES OF ACQUISITION NOISE

	$\alpha = 10$	$\alpha = 20$	$\alpha = 30$
KD	85.29%	85.20%	85.25%
DCD	87.00%	86.88%	86.83%

E. The Selection of Learning Rate

The learning rate (LR) seriously affects the student network performance by applying DCD. We compared four LR functions, including constant LR, exponential decay LR, cosine

LR, and cosine decay LR. Then empirically proposed the LR function suitable for DCD and discussed the reason.

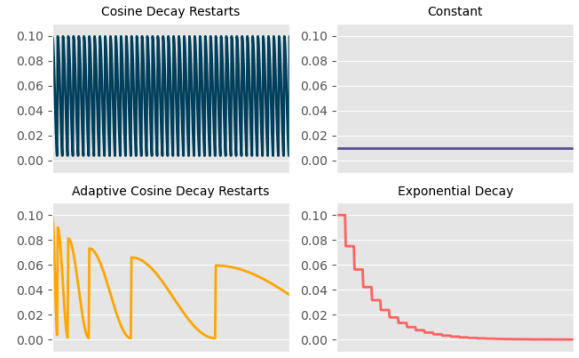


Fig. 5. The LR generated by different function in KD process

The Performance of DCD on CIFAR-100 with Various Learning Rates

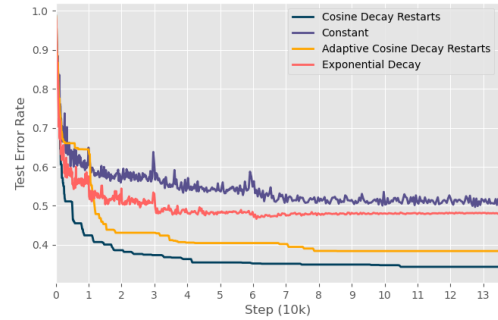


Fig. 6. The student network accuracy with different LR function

The LR generated by different function are shown in Figure5 and related convergence curves of the four LR function are shown in Figure 6. Distilling with the exponential decay or small constant learning rates is easy to fall into the local optimal in the early stage on simple samples, which damages the student network performance. Correspondingly, the cosine learning rate is more suitable for curriculum distillation because the periodic increase in learning rate helps jump out local optimal. Compared with the conventional cosine learning rate,

the adaptive cosine decay function reduces the performance of the student network. We consider that DCD provides smooth objective functions in the early stage. With complex samples added, more local optimal appear in the objective function. The small learning rate hinders the optimization process from jumping out the local optimal points in the later stage of distillation, which hurts the student network performance.

VI. CONCLUSION

In this paper, we proposed DCD - a curriculum-based distillation framework to improve student network performance. Our approach incorporates a dynamic indicator that employs the teacher and student network snapshots to measure sample complexity and a scheduler that automatically adjusts the complexity of the training set distilling process based on the student network feedback. By applying DCD, samples are fed into the distilling process in a specific order automatically until the student network convergence on the whole training set. Extensive experiments have shown that DCD significantly improves the performance of student networks not only on noiseless but also on noisy data sets.

REFERENCES

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [2] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," *arXiv preprint arXiv:1412.6550*, 2014.
- [3] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.
- [4] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, 2009.
- [5] Y. Gong, C. Liu, J. Yuan, F. Yang, and H. Wang, "Density-based dynamic curriculum learning for intent detection," 2021.
- [6] B. Zhang, Y. Wang, W. Hou, H. Wu, J. Wang, M. Okumura, and T. Shinzaki, "Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling," 2021.
- [7] Bengio, Yoshua, Courville, Aaron, Vincent, and Pascal, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [8] N. Komodakis and S. Zagoruyko, "Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer," in *ICLR*, 2017.
- [9] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4133–4141.
- [10] H. Chen, Y. Wang, C. Xu, C. Xu, and D. Tao, "Learning student networks via feature embedding," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 25–35, 2021.
- [11] C. Yang, L. Xie, S. Qiao, and A. Yuille, "Knowledge distillation in generations: More tolerant teachers educate better students," *arXiv preprint arXiv:1805.05551*, 2018.
- [12] S. Park and N. Kwak, "Feature-level ensemble knowledge distillation for aggregating knowledge from multiple networks," in *ECAI 2020*. IOS Press, 2020, pp. 1411–1418.
- [13] M. Haroush, I. Hubara, E. Hoffer, and D. Soudry, "The knowledge within: Methods for data-free model compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8494–8502.
- [14] A. Chawla, H. Yin, P. Molchanov, and J. Alvarez, "Data-free knowledge distillation for object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3289–3298.
- [15] J. Wang, L. Gou, W. Zhang, H. Yang, and H. W. Shen, "Deepvid: Deep visual interpretation and diagnosis for image classifiers via knowledge distillation," *IEEE Transactions on Visualization and Computer Graphics*, vol. PP, no. 6, pp. 1–1, 2019.
- [16] M. Zhu, K. Han, C. Zhang, J. Lin, and Y. Wang, "Low-resolution visual recognition via deep feature distillation," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.
- [17] H. Tan, X. Liu, M. Liu, B. Yin, and X. Li, "Kt-gan: Knowledge-transfer generative adversarial network for text-to-image synthesis," *IEEE Transactions on Image Processing*, vol. PP, no. 99, 2020.
- [18] K. Clark, M. T. Luong, U. Khandelwal, C. D. Manning, and Q. V. Le, "Bam! born-again multi-task networks for natural language understanding," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019.
- [19] Y. C. Chen, Z. Gan, Y. Cheng, J. Liu, and J. Liu, "Distilling knowledge learned in bert for text generation," 2019.
- [20] Z. R. Wang and J. Du, "Joint architecture and knowledge distillation in cnn for chinese text recognition," *Pattern Recognition*, vol. 111, p. 107722, 2021.
- [21] Y. Pan, F. He, and H. Yu, "A novel enhanced collaborative autoencoder with knowledge distillation for top-n recommender systems," *Neurocomputing*, vol. 332, no. MAR.7, pp. 137–148, 2019.
- [22] A. F. Perez, V. Sanguineti, P. Morerio, and V. Murino, "Audio-visual model distillation using acoustic images," in *Workshop on Applications of Computer Vision*, 2020.
- [23] L. Gao, H. Mi, B. Zhu, D. Feng, and Y. Peng, "An adversarial feature distillation method for audio classification," *IEEE Access*, vol. PP, no. 99, pp. 1–1, 2019.
- [24] Y. Zhang, P. Cong, B. Liu, W. Wang, and K. Xu, "Air: An ai-based team entry replacement scheme for routers," in *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQoS)*. IEEE, 2021, pp. 1–10.
- [25] Y. Zhang, X. Nie, J. Jiang, W. Wang, K. Xu, Y. Zhao, M. J. Reed, K. Chen, H. Wang, and G. Yao, "Bds+: An inter-datacenter data replication system with dynamic bandwidth separation," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 918–934, 2021.
- [26] Y. Zhang, P. Li, Z. Zhang, B. Bai, G. Zhang, W. Wang, B. Lian, and K. Xu, "Autosight: Distributed edge caching in short video network," *IEEE Network*, vol. 34, no. 3, pp. 194–199, 2020.
- [27] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.
- [28] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," *Advances in neural information processing systems*, vol. 23, pp. 1189–1197, 2010.
- [29] F. Ma, D. Meng, Q. Xie, Z. Li, and X. Dong, "Self-paced co-training," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2275–2284.
- [30] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2304–2313.
- [31] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 761–769.
- [32] S. Guo, W. Huang, H. Zhang, C. Zhuang, D. Dong, M. R. Scott, and D. Huang, "Curriculumnet: Weakly supervised learning from large-scale web images," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 135–150.
- [33] E. A. Platanios, O. Stretcu, G. Neubig, B. Poczos, and T. M. Mitchell, "Competence-based curriculum learning for neural machine translation," *arXiv preprint arXiv:1903.09848*, 2019.
- [34] R. El-Bouri, D. Eyre, P. Watkinson, T. Zhu, and D. Clifton, "Student-teacher curriculum learning via reinforcement learning: predicting hospital inpatient admission location," in *International Conference on Machine Learning*. PMLR, 2020, pp. 2848–2857.
- [35] C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel, "Reverse curriculum generation for reinforcement learning," in *Conference on robot learning*. PMLR, 2017, pp. 482–495.
- [36] L. N. Darlow, E. J. Crowley, A. Antoniou, and A. J. Storkey, "Cinic-10 is not imagenet or cifar-10," *arXiv preprint arXiv:1810.03505*, 2018.