Knowledge Condensation Distillation

Chenxin Li[1], Mingbao Lin[2], Zhiyuan Ding[1], Nie Lin[4], Yihong Zhuang[1],
Yue Huang[1,3⋆], Xinghao Ding[1,3], and Liujuan Cao[1]

[1] School of Informatics, Xiamen University
[2] Tencent Youtu Lab
[3] Institute of Artificial Intelligence, Xiamen University
[4] Hunan University
chenxinli@stu.xmu.edu.cn linmb001@outlook.com
dingzhiyuan@stu.xmu.edu.cn nielin@hnu.edu.cn
{zhuangyihong,yhuang2010,dxh,caoliujuan}@xmu.edu.cn

Abstract. Knowledge Distillation (KD) transfers the knowledge from a high-capacity teacher network to strengthen a smaller student. Existing methods focus on excavating the knowledge hints and transferring the whole knowledge to the student. However, the knowledge redundancy arises since the knowledge shows different values to the student at different learning stages. In this paper, we propose Knowledge Condensation Distillation (KCD). Specifically, the knowledge value on each sample is dynamically estimated, based on which an Expectation-Maximization (EM) framework is forged to iteratively condense a compact knowledge set from the teacher to guide the student learning. Our approach is easy to build on top of the off-the-shelf KD methods, with no extra training parameters and negligible computation overhead. Thus, it presents one new perspective for KD, in which the student that actively identifies teacher's knowledge in line with its aptitude can learn to learn more effectively and efficiently. Experiments on standard benchmarks manifest that the proposed KCD can well boost the performance of student model with even higher distillation efficiency. Code is available at https://github.com/dzy3/KCD.

Keywords: Knowledge distillation; Active learning; Efficient training

1 Introduction

Though deep neural networks (DNNs) have achieved great success in computer vision, most advanced models are too computationally expensive to be deployed on the resource-constrained devices. To address this, the light-weight DNNs have been explored in the past decades. Typical methods include network pruning [18], parameter quantization [36] and neural architecture search [2], *etc.* Among all these methods, knowledge distillation [10] is widely integrated into their learning frameworks, whereby the original cumbersome model (teacher) transfers its

⋆ Corresponding Author

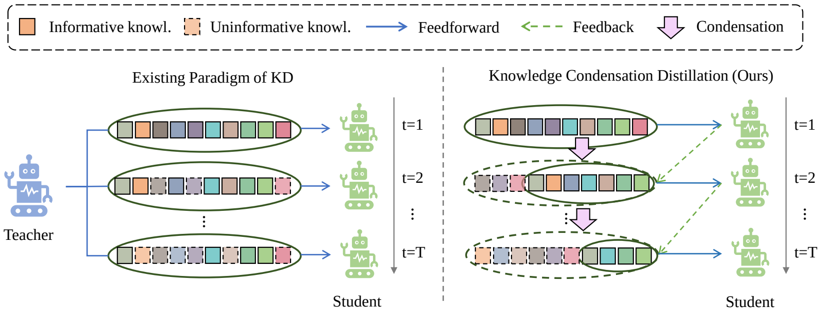


Fig. 1: Comparison between existing KD paradigm and our KCD. **Left**: Existing paradigm transfers the complete knowledge points from teacher model across the entire training process, regardless of the varying values to the student at different stages. **Right**: Knowledge points are first estimated based on the current capacity of the student, and then condensed to a compact yet informative sub-part for student model.

knowledge to enhance the recognition capacity of its compressed version, *a.k.a.* student model. Due to its flexibility, KD has received ever-increasing popularity in varieties of vision tasks.

In most existing studies on KD [10,25,13,30,22,4,17,16], the knowledge hints of the whole sample space, such as soft predictions [10], intermediate representations [25], attention maps [13], *etc*, are transferred to the student model across the entire training process as illustrated in Fig. 1(**Left**). However, these methods neglect the changing capacity of the student model at different learning stages. Specifically, all the knowledge points from teacher model are informative enough for the student model at its infant learning stage. However, as the learning proceeds, the value of different knowledge points starts to vary for the student. For example, the "well-memorized" knowledge points have a relatively limited impact to the student at the later training stages. Consequently, the concern regarding redundancy of knowledge transfer arises in existing studies, whereby the student model passively receives all the knowledge points from the teacher. This further poses two severe issues: **(1)** Training burden. The redundant knowledge requires not only additional memory storage, but also prolongs the training time. **(2)** Poor performance. The redundancy prevents the student model from concentrating enough on the more informative knowledge, which weakens the learning efficacy of the student model.

To overcome the above challenge, as shown in Fig. 1(**Right**), this paper presents a new perspective for KD, the cores of which are two main folds: **(1)** A feedback mechanism is introduced to excavate various values of the teacher's knowledge for the student at different training stages. **(2)** The student actively identifies the informative knowledge points and progressively condenses a core knowledge set for distillation. To this end, we propose a Knowledge Condensation Distillation (KCD) paradigm, whereby the label of knowledge value to student model is encoded as a latent variable, and an Expectation-Maximization (EM) framework is forged to iteratively condense the teacher's knowledge set and distill the student model. Furthermore, given the local-batch training fashion in student learning, we propose an Online Global Value Estimation (OGVE) module to dynamically estimate the knowledge value over the global knowledge space. To generate a compact yet effective encoding of teacher's knowledge, we develop a Value-Adaptive Knowledge Summary (VAKS) module to adaptively preserve high-value knowledge points, remove the valueless points as well as augment the intermediate ones. We conduct extensive experiments on two benchmarks, CIFAR100 [14] and ImagetNet [5], and many representative settings of teacher-student networks in KD. We show that our KCD can be well built upon majorities of existing KD pipelines as a plug-and-play solution, without bringing extra training parameters and computation overhead.

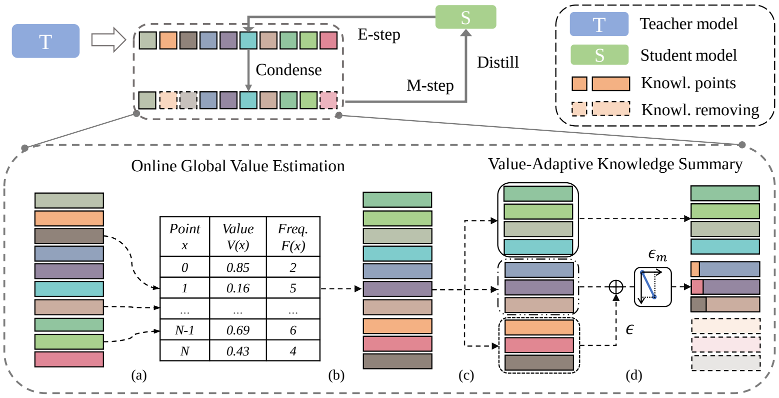
Our contributions are summarized as follows:

- We propose a novel KD paradigm of knowledge condensation, wherein the knowledge to transfer is actively determined by the student model, and a concise encoding condensed from the whole knowledge set is utilized in KD.
- We derive an Expectation-Maximization framework to accomplish our knowledge condensation distillation by iteratively performing knowledge condensation and model distillation.
- We propose an OGVE module to acquire an approximate global estimator of knowledge value while utilizing only local training statistics. We further present a VAKS module to harmonize the trade-off between compactness and informativeness of knowledge condensation encoding.

2 Related Work

Knowledge Distillation. The pioneering KD work dates back to [10], where the soft probability distribution from the teacher is distilled to facilitate the student's training. Since then, abundant developments have been committed to excavating richer knowledge hints, such as intermediate representation [25,9], attention maps [13], instance relation [32,23], self-supervised embedding [30,34] and so on. All these methods transfer the knowledge on all training instances to the student regardless of different training stages. Differently, we study the redundancy of the teacher's knowledge and emphasize the significance of making the student model actively condense an efficient knowledge set for learning.

One more recent study [35] considers the efficiency issue of KD by identifying the most informative samples in each training batch. Our method differs from the following aspects. First, the study [35] explores the difference of computation overheads in the forward passes of the teacher and student models, and fixes the knowledge set during the distillation process. As a comparison in our method, the knowledge set is dynamically condensed and explicitly encodes the patterns of the student model during training. Second, we estimate the knowledge value over the complete sample space rather than every single batch, which is more accurate and comprehensive.



Point x	Value V(x)	Freq. F(x)
0	0.85	2
1	0.16	5
...	...	...
N-1	0.69	6
N	0.43	4

(a) Online global recording (b) Cost-aware value ranking (c) Knowl. partition (d) Adaptive knowl. augmentation

Fig. 2: Overview of the proposed KCD framework. The knowledge condensation and student distillation are optimized iteratively in an EM framework.

Coreset Construction. Another related literature is the problem of coreset construction [8,26]. The main idea behind them is that a learning agent can still perform well with fewer training samples by selecting data itself. Most existing works [12,33,31,20,40] construct this coreset by importance sampling. For example, In [12], sample importance is estimated via the magnitude of its loss gradient *w.r.t.* model parameters. CRAIG [20] selects a weighted coreset of training data that closely estimates the full gradient by maximizing a submodular function. Wang *et al.* [33] distilled the knowledge from the entire dataset to generate a synthetic smaller one. These ideas inspire us to seek a core component of the whole knowledge set from the teacher to realize an efficient KD.

3 Methodology

3.1 Preliminaries

In the task of knowledge distillation (KD), we are given a training dataset $\mathcal{X}$, a pre-trained teacher model $\mathcal{T}$ and a to-be-learned student model $\mathcal{S}$. Hinton *et al.* [10] proposed to minimize the cross-entropy loss between the output probability $p_\mathcal{T}(x)$ of the teacher and that $p_\mathcal{S}(x)$ of the student:

$$\mathcal{L}_{KD} = -\sum_{x\in\mathcal{X}} p_\mathcal{T}(x) \log\big(p_\mathcal{S}(x)\big). \tag{1}$$

Denoting each pair $\big(x, p_\mathcal{T}(x)\big)$ as a knowledge point, the teacher $\mathcal{T}$ in essence provides a knowledge set $K = \{\big(x, p_\mathcal{T}(x)\big)|x \in \mathcal{X}\}$, which is then transferred to the student $\mathcal{S}$. In conventional KD, the knowledge set K is fixed across the whole distillation process, despite different learning stages of the student model. As a core distinction, we propose to transfer simply a concise knowledge encoding $\hat{K}$

with $|\hat{K}| < |K|$, where the knowledge points are most valuable and adapt to the demand of the student model at different periods.

In what follows, we show that the efficient coding $\hat{K}$ can be deduced by the Expectation-Maximization (EM) framework that encodes the knowledge value for the student model as a latent variable Y, with which, we can identify the most valuable components in knowledge set K. Fig. 2 shows the overview of the proposed method.

3.2 Knowledge Condensation Distillation

The goal of KD in Eq. (1) is to learn the parameter θ of student model in order to maximize the negative cross-entropy between the teacher $\mathcal{T}$ and student $\mathcal{S}$:

$$\hat{\theta} = \arg\max_{\theta} \sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} p_{\mathcal{T}}(x, c) \log p_{\mathcal{S}}(x, c; \theta), \tag{2}$$

where $\mathcal{C}$ denotes the class space. Instead of transferring the complete knowledge set $K = \{\big(x, p_{\mathcal{T}}(x)\big) | x \in \mathcal{X}\}$, we introduce a binary value variable $\mathcal{Y} \in \{0, 1\}^{|K|}$, the i-th value of which indicates if the i-th knowledge point is valuable to the student. In this way, the traditional optimization of Eq. (2) in our setting becomes:

$$\hat{\theta} = \arg\max_{\theta} \sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} p_{\mathcal{T}}(x, c) \log \sum_{y \in \mathcal{Y}} p_{\mathcal{S}}(x, c, y; \theta). \tag{3}$$

To maximize this objective, we consider its low-bound surrogate:

$$\begin{aligned} & \sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} p_{\mathcal{T}}(x, c) \log \sum_{y \in \mathcal{Y}} p_{\mathcal{S}}(x, c, y; \theta) \\ = & \sum_{x \in \mathcal{X}} \sum_{x \in \mathcal{C}} p_{\mathcal{T}}(x, c) \log \sum_{y \in \mathcal{Y}} Q(y) \frac{p_{\mathcal{S}}(x, c, y; \theta)}{Q(y)} \\ \geq & \sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} p_{\mathcal{T}}(x, c) \sum_{y \in \mathcal{Y}} Q(y) \log \frac{p_{\mathcal{S}}(x, c, y; \theta)}{Q(y)}, \end{aligned} \tag{4}$$

where $Q(y)$ denotes the distribution over the space of value labels $\mathcal{Y}$ so that $\sum_{y \in \mathcal{Y}} Q(y) = 1$. Note that we derive the last step based on Jensen's inequality where the equality holds if and only if $\frac{p_{\mathcal{S}}(x,c,y;\theta)}{Q(y)}$ is a constant [15]. Under this condition, the distribution $Q(y)$ should be:

$$Q(y) = \frac{p_{\mathcal{S}}(x, c, y; \theta)}{\sum_{y \in \mathcal{Y}} p_{\mathcal{S}}(x, c, y; \theta)} = \frac{p_{\mathcal{S}}(x, c, y; \theta)}{p_{\mathcal{S}}(x, c; \theta)} = p_{\mathcal{S}}(y; x, c, \theta). \tag{5}$$

Removing the constant term $-\sum_{y \in \mathcal{Y}} Q(y) \log Q(y)$ in Eq. (4) and combining Eq. (5) lead to our final optimization:

$$\sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} p_{\mathcal{T}}(x, c) \sum_{y \in \mathcal{Y}} p_{\mathcal{S}}(y; x, c, \theta) \log p_{\mathcal{S}}(x, c, y; \theta). \tag{6}$$

The maximization of the above problem can be realized by Expectation-Maximization (EM) algorithm, as elaborated below:

E-step. In this step, we aim to evaluate value distribution $Q(y) = p_{\mathcal{S}}(y; x, c, \theta)$. Before that, we first discuss how to measure the value of each knowledge point $(x, p_{\mathcal{T}(x)})$, insight of which is two-fold: First, it has been verified that the average prediction entropy loss decreases drastically if a model is distilled by knowledge hints from a teacher model, instead of being trained solely [21]. This reflects the contribution of knowledge points to the training of the student model. Second, as discussed in [27], the knowledge which encodes informative semantic structure tends to require more training time for the student model to fit well.

These two insights indicate that the prediction entropy loss can be an option to measure the knowledge value. Besides, informative knowledge tends to cause a larger entropy loss. Therefore, given a knowledge point $(x, p_{\mathcal{T}(x)})$, we utilize its prediction entropy to measure its value:

$$V(x) = -\sum_{c \in \mathcal{C}} p_{\mathcal{S}}(x, c) \log p_{\mathcal{S}}(x, c). \tag{7}$$

With the prediction entropy, in order to estimate $p_{\mathcal{S}}(y; x, c, \theta)$, we further conduct a ranking operation in an decreasing order *w.r.t.* $V(x)$ over $\mathcal{X}$. Then, based on the ranking position $\mathcal{R}_V(x) \in \{0, 1, \cdots, N\}$, we derive the relative likelihood probability about knowledge value:

$$p_{\mathcal{R}_V}(y; x, \theta) = 1 - \frac{\mathcal{R}_V(x)}{|\mathcal{X}|}. \tag{8}$$

Then the likelihood of value label $p_{\mathcal{S}}(y; x, c, \theta)$ can be determined by a threshold τ: $p_{\mathcal{S}}(y; x, c, \theta) = 1$ if $p_{\mathcal{R}_V}(y; x, \theta) \geq \tau$, and 0 otherwise.

M-step. After E-step, the optimized object of Eq. (6) can be re-written as:

$$\begin{aligned} &\sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} p_{\mathcal{T}}(x, c) \sum_{y \in \mathcal{Y}} p_{\mathcal{S}}(y; x, c, \theta) \log p_{\mathcal{S}}(x, c, y; \theta) \\ = &\sum_{x \in \mathcal{X}} \sum_{c \in \mathcal{C}} p_{\mathcal{T}}(x, c) \sum_{y \in \mathcal{Y}} \mathbb{I}(p_{\mathcal{R}_V}(y; x, \theta) \geq \tau) \log p_{\mathcal{S}}(x, c, y; \theta), \end{aligned} \tag{9}$$

where $\mathbb{I}(\cdot)$ returns 1 if its input is true, and 0 otherwise. When the training samples are not provided, we assume a uniform priori over y (0 or 1):

$$p_{\mathcal{S}}(x, c, y; \theta) = p_{\mathcal{S}}(x, c; y, \theta) p_{\mathcal{S}}(y; \theta) = \frac{1}{2} \cdot p_{\mathcal{S}}(x, c; y, \theta), \tag{10}$$

where $p_{\mathcal{S}}(y; \theta) = \frac{1}{2}$ due to the premise of uniform distribution. Then the distribution $p_{\mathcal{S}}(x, c; y, \theta)$ is only conditioned on the estimated value label Y, *i.e.*, $\mathbb{I}(p_{\mathcal{R}_V}(y; x, \theta) \geq \tau)$. We conduct distillation only upon the knowledge points with label $y = 1$. Thus, we can re-write the maximum estimation in Eq. (2) as:

$$\hat{\theta} = \arg\min_{\theta} \sum_{x \in \mathcal{X} | Y(x) = 1} \sum_{c \in \mathcal{C}} -p_{\mathcal{T}}(x, c) \log p_{\mathcal{S}}(x, c; \theta), \tag{11}$$

where $x \in \mathcal{X}$ can be used for distillation only when the condition $y = 1$.

Consequently, our KCD iteratively performs E-step and M-step. The former aims to find the distribution of label Y and the concise knowledge encoding $\hat{K}$ comprises these knowledge points with $y = 1$, while the latter implements efficient distillation upon the concise set $\hat{K}$. However, current neural networks are trained in a batch fashion where a small portion of samples are fed forward each time. These local sample batches barricade a direct extraction of the concise knowledge set $\hat{K}$ from the whole training dataset $\mathcal{X}$. In what follows, we further propose an online global value estimation to solve this problem.

3.3 Online Global Value Estimation

In order to condense a valuable knowledge set $\hat{K}$ in a global fashion, we design an online global value estimation (OGVE) to derive the global statistics of the whole training dataset $\mathcal{X}$ which consists of an online value recording and cost-aware ranking below:

Online Value Recording. The estimation of $p_{\mathcal{S}}(y; x, c, \theta)$ is conducted by $p_{\mathcal{R}_V}(y; x, \theta)$ over the whole (global) sample space $\mathcal{X}$ at the E-step. However, only a small-portion sub-set (local) of knowledge can be available at each training iteration. Besides, the same sample x might appear frequently at different training stages. To alleviate this issue, we propose to consider the historical statistics of x. Technically, when x is fed to the network at a particular training iteration, we first count the frequency of x ever involving in the training, denoted as $F(x)$. Also, we calculate its prediction entropy $V(x)$ at current training iteration using Eq. (7). Then, the global value of a knowledge point $\big(x, p_{\mathcal{T}(x)}\big)$ is updated in an online moving-average fashion as:

$$V^{F(x)}(x) = \frac{F(x) - 1}{F(x)} \times V^{F(x)-1} + \frac{1}{F(x)} V(x). \tag{12}$$

Cost-Aware Ranking. Based on the recorded global statistics of $V^{F(x)}(x)$, we can obtain a more accurate ranking of $R_V(x)$ without introducing any additional overhead. However, in the current design, the ranking order of two knowledge points with a similar value might be the same even though their training frequencies are very different, which is counter-intuitive, given the fact that the neural network tends to memorize and gives low prediction entropy to these samples that have ever seen more times [7]. Therefore, the knowledge points with similar $V(x)$ but a higher training cost $F(x)$ should be more critical for the student model and assigned with a top ranking. Considering this, in the ranking operation, we re-weight $V(x)$ using the training frequency $F(x)$ as:

$$\mathcal{R}_V(x) = \underset{x \in \mathcal{X}}{\arg\,\mathrm{sort}}\ V^{F(x)}(x) \times \big(F(x)\big)^{\alpha}, \tag{13}$$

where α controls the weighted effect of $F(x)$. Eq. (13) considers not only the status of $V(x)$, but also the cost $F(x)$ to achieve this status.

Combining $\mathcal{R}_V(x)$ in Eq. (13) and $p_{\mathcal{R}_V}(y; x, \theta)$ in Eq. (8), we can estimate the value label Y. Accordingly, the concise knowledge encoding $\hat{K}$ consists of the knowledge points with $y = 1$. As the training proceeds, many well learned knowledge points becomes less valuable to the student model. However, the relative likelihood probability $p_{\mathcal{R}_V}(y; x, \theta) \neq 0$ indicates a possibility to be selected again. Instead, we further propose a value-adaptive knowledge summary by solving this issue in a divide-and-conquer manner.

3.4 Value-Adaptive Knowledge Summary

Our value-adaptive knowledge summary (VAKS) performs concise knowledge encoding in a two-step fashion including a knowledge partition and an adaptive knowledge augmentation.

Knowledge Partition. According to our OGVE, we can obtain an explicit label set Y. Then, the original knowledge set can be divided into K_1 with $y = 1$ and K_0 with $y = 0$. For knowledge points in K_0, they are deemed to be valueless thus we choose to directly discard them. As for K_1, based on the relative likelihood probability $\mathcal{R}_V(x)$, we further partition it into a set K_{1H}, element of which has a relatively high $\mathcal{R}_V(x)$, and a set K_{1L}, element of which has relatively low $\mathcal{R}_V(x)$, as shown in Fig. 2. Besides, our partition also requires K_{1L} to be in the same size with K_0, *i.e.*, $|K_{1L}| = |K_0|$, reason of which will be given in the following adaptive knowledge augmentation.

The knowledge points in K_{1H} are of high possibility to be valuable for the student, thus they can be safely transferred to the student as the conventional KD does. However, the knowledge in K_{1L} falls into a "boundary status". Though considered valuable, they are prone to being less valuable than knowledge in K_{1H} and easily absorbed by the student. This motivates us to enhance the knowledge points in K_{1L}. One straightforward approach is to introduce the gradient-based distillation [33,37,41] to generate new knowledge contents. However, the heavy time consumption barricades its application. In what follows, we introduce an adaptive knowledge augmentation to reach this goal in a training-free fashion.

Adaptive Knowledge Augmentation. Our insight of knowledge augmentation comes from the field of adversarial examples [29,6], where a subtle perturbation can greatly confuse the model recognition. Likewise, we also seek a perturbation on the knowledge points in K_{1L}. It is noteworthy that rather than to find the most disruptive disturbance in adversarial examples, our goal is to use some knowledge-wise perturbation to augment the knowledge points, making them more informative for the student model.

Concretely, denoting $S = \{|K_1|, |K_1| - 1, ..., |K_{1L}|\}$, as shown in Fig. 2, we propose to make full use of the removed valueless knowledge in K_0 to augment knowledge points in K_{1L} with a very small perturbation ratio ϵ. as:

$$K_{Aug} = \text{Ordered}(K_{1L}) \oplus \text{Ordered}(K_0) \otimes \epsilon(S), \tag{14}$$

where Ordered$(\cdot)$ reorders its input set in descending according to the value of knowledge point, and $\oplus$ denotes the element-wise adding operation. Recall that $|K_{1L}| = |K_0|$ in our setting, thus, the $\oplus$ is applicable. $\epsilon(\cdot)$ is defined as:

$$\epsilon(x') = \frac{\epsilon_m}{|K_0|}(x' - |K_1|) + \epsilon_m. \tag{15}$$

```
Algorithm 1 Knowledge Condensation Distillation
Input: Training dataset $\mathcal{X}$; a student model $\mathcal{S}$ with learnable parameters $\theta$; a full knowledge set $K$ generated by a pre-trained teacher model $\mathcal{T}$.
Required: Number of epochs in a learning stage $T$; Desired final knowledge condensation ratio $\rho$.
Output: Distilled student model with parameter $\hat{\theta}$; condensed knowledge encoding $\hat{K}$ ($|\hat{K}| = |K| \cdot \rho$).
Init. $\hat{K} = K$;
for $i = 0, ..., I$ epoch do
    # M-step: Knowledge distillation
    Distill $\hat{\theta}$ of student $\mathcal{S}$ on the condensed knowledge $\hat{K}$ via Eq. (11);
    #E-step: Knowledge condensation
    ## Estimate knowledge value over $K$ via proposed OGIE (Sec. 3.3)
    Cal. knowledge value $V(x)$ over compact (local) knowledge space $\hat{K}$ via Eq. (7);
    online update historical recording $V^{F(x)}(x)$ over complete (global) knowledge space $K$ via Eq. (12);
    if $i \ \% \ T = 0$ then
        Cal. ranking position of knowledge value $\mathcal{R}_V(x)$ via Eq. (13); cal. ranking-based likelihood probability $p_{\mathcal{R}_V}(x)$ via Eq. (8);
        Binarize $p_{\mathcal{R}_V}(y;x)$ with the threshold $\tau^t$ at current stage $t = i/T$; determine value label $Y$ ($y = 1$ or $0$) over complete knowledge space $K$;
        ## Summarize knowledge encoding $\hat{K}$ via proposed VAKS (Sec. 3.4);
        Partition $K$ into $K_1$ and $K_0$ via label $Y$; partition $K_1$ into $K_{1H}$ and $K_{1L}$, s.t. $|K_{1L}| = |K_0|$;
        Augment $K_{1L}$ via Eq. (14) and Eq. (15);
        Summarize compact knowledge encoding via $\hat{K} = K_{1H} \cup K_{Aug}$.
    end if
end for
```

Thus $\epsilon(S)$ is a set linearly increasing from 0 to a pre-given ϵ_m (see Fig. 2). The intuition of $\epsilon(x')$ is to make the knowledge points with lower-ranking positions *w.r.t.* knowledge value to get more augmentation effect while the ones with higher positions to maintain more their original knowledge contents.

Finally, we obtain the knowledge condensation $\hat{K} = K_{1H} \cup K_{Aug}$.

3.5 Overall Procedure

The overall procedure of our proposed KCD is depicted in Alg. 1. The proposed framework iteratively performs knowledge condensation in E-step and knowledge distillation in M-step, which can be practically formulated as a stage-based learning framework. The total I training epochs are equally divided into I/T learning stages, each with T epochs. Within each stage, the distillation is conducted for T epochs on the fixed knowledge set, followed by the recording of knowledge value in every training batch (Eq. (12)). At the end of each stage, we perform a ranking step across the whole knowledge set *w.r.t.* knowledge value (Eq. (13)) and knowledge summary (Eq. (14)). to condense a smaller informative knowledge set. The condensed one is then used in the next stage.

It is noteworthy that the reduction of computation overhead mainly comes from using more compact knowledge encoding $\hat{K}$ during KD. To quantitatively portray this, absolute cost C_a is calculated by the number of knowledge points used, *e.g.*, $C_a = |K| \cdot I$ for conventional KD. We further calculate the relative cost C as the rate of C_a between our KCD and the conventional KD baseline:

$$C = \frac{|K| \cdot (\tau^0 + \tau^1 + \cdots + \tau^t + \cdots + \tau^{I/T}) \cdot T}{|K| \cdot I} \tag{16}$$

where τ^t denotes the threshold of the ranking-based probability $p_{\mathcal{R}_V}$ (Eq. (8)) for the value label Y at the t-th stage. It controls the condensation ratio, as $|\hat{K}| = |K| \cdot \tau^t$ at t-th stage and the final condensation rate $\rho = \tau^{I/T}$.

4 Experiments

Datasets. We conduct experiments on two benchmark datasets for KD, namely CIFAR100 [14] and ImageNet [5]. CIFAR100 contains 50K training images with 500 images per class and 10K test images with 100 images per class. The image size is 32×32. ImageNet is a large-scale classification dataset, containing 1.2 million images over 1K classes for training and 50K for validation. The image size is 224×224.

Implementation Details. Following the common practice in [30,34], we adopt the stochastic gradient descent (SGD) optimizer with a momentum of 0.9, weight decay of 5×10^{-4}. Batch size is set as 64 for CIFAR-100 and 256 for ImageNet. For CIFAR100 [14], the learning rate is initialized as 0.05, and decayed by 0.1 every 30 epochs after the first 150 epochs until the last 240 epochs. For ImageNet [5], the learning rate is initialized as 0.1, and decayed by 0.1 every 30 epochs. Without specification, the hyper-parameters in Alg. 1 is set as follows: We set $I = 240$, $T = 40$ for CIFAR100 and $I = 90$, $T = 15$ for ImageNet. We set final condensation rate $\rho = 0.7$. The intermediate value of condensation threshold τ is set as exponential decay after every learning stage, with the initial value of $\tau^0 = \sqrt[T/E]{\rho} = 0.9423$. We set $\alpha = 0.03$ in Eq. (13), and the perturbation rate ϵ in Eq. (15) as linear growth from 0 to 0.3 ($\epsilon_m = 0.3$).

4.1 Comparisons with State-of-the Arts

Results on CIFAR100. We make comparison to various representative state-of-the-art KD methods, including vanilla KD [10], FitNet [25], AT [13], SP [32], VID [1], RKD [23], PKT [24], CRD [30], WCoRD [3], ReviewKD [4], SSKD [34]. We directly cite the quantitative results reported in their papers [30,4,3,22]. For the network of teacher and student models, we use Wide residual networks [38] (abbreviated as WRNd-w), MobileNetV2 [11] (MN2), ShuffleNetV1 [39] /ShuffleNetV2 [19] (SN1/SN2), and VGG13/VGG8 [28] (V13/V8). R110, R56 and R20 denote CIFAR-style residual networks, while R50 denotes an ImageNet-style ResNet50. **Teacher** and **Student** stand for the performance of teacher and student models when they are trained individually.

Table 1: Test Acc. (%) of the student networks on CIFAR100. **Bold** and underline denote the best and second best results. The comparison of whether to equip modern methods with our KCD is provided as (+/-). Same-architecture and cross-architecture experiments are shown in two groups of columns.

Teacher Student	W40-2 W16-2	W40-2 W40-1	R56 R20	R32x4 R8x4	V13 V8	V13 MN2	R50 MN2	R50 V8	R32x4 SN1	R32x4 SN2	W40-2 SN1
Teacher Student	75.61 73.26	75.61 73.26	72.34 69.06	79.42 72.50	74.64 70.36	74.64 64.60	79.34 64.60	79.34 70.36	79.42 70.50	79.42 71.82	75.61 70.50
KD [10]	74.92	73.54	70.66	73.33	72.98	67.37	67.35	73.81	74.07	74.45	74.83
FitNet [25]	73.58	72.24	69.21	73.50	71.02	64.14	63.16	70.69	73.59	73.54	73.73
AT [13]	74.08	72.77	70.55	73.44	71.43	59.40	58.58	71.84	71.73	72.73	73.32
SP [32]	73.83	72.43	69.67	72.94	72.68	66.30	68.08	73.34	73.48	74.56	74.52
VID [1]	74.11	73.30	70.38	73.09	71.23	65.56	67.57	70.30	73.38	73.40	73.61
RKD [23]	73.35	72.22	69.61	71.90	71.48	64.52	64.43	71.50	72.28	73.21	72.21
PKT [24]	74.54	73.45	70.34	73.64	72.88	67.13	66.52	73.01	74.10	74.69	73.89
CRD [30]	75.64	74.38	71.63	75.46	74.29	69.94	69.54	74.58	75.12	76.05	76.27
WCoRD [3]	76.11	74.72	<u>71.92</u>	<u>76.15</u>	74.72	70.02	70.12	74.68	75.77	76.48	76.68
ReviewKD [4]	<u>76.12</u>	75.09	71.89	75.63	74.84	70.37	69.89	-	77.45	77.78	<u>77.14</u>
SSKD [34]	75.66	<u>75.27</u>	70.96	75.80	**75.12**	<u>70.92</u>	<u>71.14</u>	**75.72**	<u>77.91</u>	<u>78.37</u>	76.92
KC-KD	75.70 (+0.78)	73.84 (+0.30)	70.75 (+0.09)	74.05 (+0.72)	73.44 (+0.46)	68.61 (+1.24)	67.94 (+0.59)	74.41 (+0.60)	74.33 (+0.26)	75.19 (+0.74)	75.60 (+0.77)
KC-PKT	75.01 (+0.47)	74.12 (+0.67)	72.08 (+1.74)	74.45 (+0.81)	72.82 (-0.06)	67.99 (+0.86)	67.92 (+1.40)	73.32 (+0.31)	74.60 (+0.50)	75.79 (+1.10)	75.78 (+1.89)
KC-CRD	75.93 (+0.29)	74.60 (+0.22)	**72.11** (+0.48)	75.78 (+0.32)	74.38 (+0.09)	69.90 (-0.04)	69.82 (+0.28)	74.49 (-0.09)	75.74 (+0.62)	76.44 (+0.39)	76.40 (+0.13)
KC-SSKD	**76.24** (+0.58)	**75.35** (+0.08)	71.31 (+0.35)	**76.48** (+0.68)	<u>74.93</u> (-0.21)	**71.32** (+0.40)	**71.29** (+0.15)	<u>75.65</u> (-0.07)	**78.28** (+0.37)	**78.59** (+0.22)	**77.61** (+0.69)

Table 2: Test Acc. (%) with computation cost C on CIFAR100, compared with the only existing method UNIX [35] that focuses on distillation efficiency.

Teacher Student	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	resnet56 resnet20	VGG13 VGG8	VGG13 MobileNetV2	ResNet50 VGG8
KD	74.92 (100%)	73.54 (100%)	70.66 (100%)	72.98 (100%)	67.37 (100%)	73.81 (100%)
UNIX-KD	75.19 (75.3%)	73.51 (73.1%)	70.06 (76.0%)	73.18 (76.4%)	68.47 (77.5%)	73.62 (68.9%)
UNIX-KD†	75.25 (81.6%)	**74.18** (81.6%))	70.19 (81.6%)	73.27 (81.6%)	68.58 (81.6%)	74.24 (81.6%)
KC-KD	**75.70** (81.6%)	73.84 (81.6%)	**70.75** (81.6%)	**73.44** (81.6%)	**68.61** (81.6%)	**74.41** (81.6%)

The experimental results on 11 teacher-student pairs are depicted in Tab. 1. We can see that constructing the proposed knowledge condensation (KC) on vanilla KD shows an impressive improvement. In addition, our KCD on top of various modern KD approaches all demonstrates an obvious accuracy gain. More importantly, the proposed KCD utilizes only the condensed knowledge, which enjoys the merits of both accuracy and efficiency.

We also compare KCD with the only existing work that focuses on the computation cost of KD, namely UNIX [35]. Tab. 2 displays the results on accuracy and computation cost C[5] (see Eq. (16)). It is noteworthy that the computation C of our KCD is unrelated to the network, thus keeps fixed across different teacher-

[5] Calculation process of C of our method is detailed in supplementary materials.

Table 3: Top-1/-5 error (%) on ImageNet, from ResNet34 to ResNet18. Equipping our method reduces computation cost C:100%→81.61%, C_a:114M→81M.

	Tea.	Stu.	AT [13]	SP [32]	OnlineKD [42]	SSKD [34]	KD [10]	KC-KD (Ours)	CRD [30]	KC-CRD (Ours)	ReKD [4]	KC-ReKD (Ours)
Top-1	26.69	30.25	29.30	29.38	29.45	28.38	29.34	28.61	28.83	28.46	28.39	**27.87**
Top-5	8.58	10.93	10.00	10.20	10.41	9.33	10.12	9.62	9.87	9.53	9.49	**9.08**

Table 4: Ablation study of the proposed KCD on four KD processes (%).

Teacher	WRN-40-2	VGG13	VGG13	resnet32x4
Student	WRN-16-2	VGG8	MobileNetV2	ShuffleNetV2
OGVE w/ random	74.54	73.01	67.76	74.92
OGVE w/o OVR	75.01	73.26	67.73	74.56
OGVE w/o CAR	75.27	73.04	67.79	74.88
OGVE -Full	75.48	73.08	68.23	75.16
OGVE + VAKS w/ KA ($\epsilon = \epsilon_m$)	75.57	73.20	68.34	75.14
OGVE + VAKS (Full)	**75.70**	**73.44**	**68.61**	**75.19**

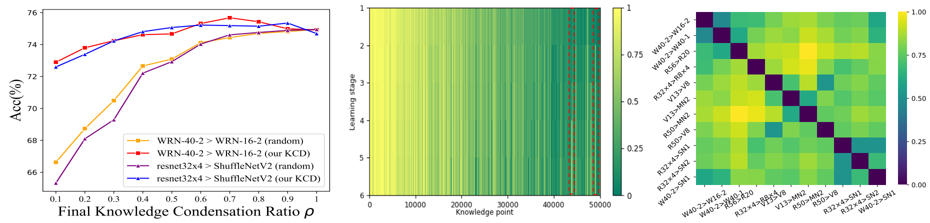
student pairs. In contrast, C in UNIX [35] is dependent on the rate among the number of sample pass in teacher forward, student forward, and student backward, thus it presents diverse values across different pairs. KD denotes the vanilla baseline, with C set as 100%. UNIX denotes citing the accuracy results of the models reported in the original works which have the most similar C with our KCD. UNIX† denotes using their public code[6] to run and evaluate their methods with the same cost setting C with our method. It appears that the accuracy of the proposed KCD outperforms UNIX at the same level of computation cost.

Results on ImageNet. Following common practice [30,34], the experiments on ImageNet are conducted using ResNet34 (teacher) and ResNet18 (student). Tab. 3 displays the results of both Top-1 and Top-5 error. We can see that building the proposed knowledge condensation upon KD, CRD and ReviewKD (abbreviated as ReKD) all reduces the testing error significantly. Moreover, the proposed KC leads to the reduction of relative computation C to 81.61%, and the absolute computation C_a from 114M to 81M, which reveals an obvious gain of training efficiency in the large-scale benchmark.

4.2 Further Empirical Analysis

Ablation Study. We verify the effect of each component in the proposed framework by conducting ablation studies. The results are provided in Tab. 4. **(1)** OGVE w/ random denotes we randomly allocate ranking position $\mathcal{R}_V$ as well as value label Y instead of using OGVE. **(2)** OGVE w/o OVR, w/o CAR, -full denotes that we remove online value recording (*i.e.*, estimate the value during

[6] https://github.com/xuguodong03/UNIXKD



(a) Accuracy with ratio ρ (b) Varying knowledge value (c) Distance matrix

Fig. 3: (a) Accuracy with the variation of condensation ratio ρ. (b) Pattern of varying knowledge value across the training process. (c) Hamming Distance matrix of the value label across KD processes.

the mini-batch training), cost-aware ranking (*i.e.*, discard the weight $\big(F(x)\big)^{\alpha}$ in Eq. (13), and keep the full setting of OGVE. Note that the above variants of OGVE are combined with a direct selection on label y=1. **(3)** OGVE + VAKS w/ KA (ϵ=ϵ_m) denotes using non-adaptive knowledge augmentation with ϵ keeping its maximum ϵ_m=0.3 for all points. OGVE + VAKS denotes the full structure of our proposed KCD. When any component is removed, it appears that the performance drops accordingly, revealing the effectiveness of our design.

Influence of Knowledge Condensation Ratio ρ. Fig. 3(a) displays the performance of different models (*i.e.*, random selection baseline and our KCD) on different KD processes (*i.e.*, W40-2>W16-2 and R32×4>SN2), with the variation of final knowledge condensation ratio ρ. We can see that our KCD outperforms the random baseline by a significant margin, especially as ρ decreasing. It is noteworthy that the proposed KCD achieves better results on ρ ranging between 0.6-0.8 than full-knowledge setting with ρ=1, and nearly maintains the accuracy among a wide range ρ as 0.3-1.0, which implies that our KCD can identify and summarize the compact yet effective knowledge encoding that is robust to the size reduction of knowledge set.

Pattern of Knowledge Value. Fig. 3(b) displays the varying ranking-based probability *w.r.t* knowledge value during the entire training process. We can see that the value of knowledge points differs to the student at different learning stages. As marked in red, some knowledge points are valueless at the beginning stage while become more and more critical in the later stages. Fig. 3(c) depicts the Hamming distance matrix about the estimated value label Y in the final stage across various KD processes, wherein the distance indicates the number of different elements in two masks. We can see that the value label represents a relatively strong correlation (small distance) when two KD processes have the same student architectures (*e.g.*, V13>V8 and R50>V8) or similar ones (*e.g.*, R32×4>SN1 and R32×4>SN2), revealing that the identified knowledge value really encodes some "patterns" of the student model.

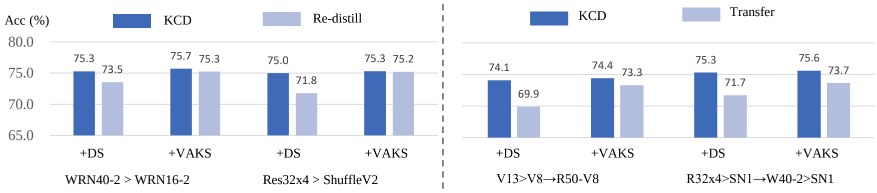


Fig. 4: The performance of reusing the condensed knowledge. **Left**: We utilize the condensed knowledge to directly re-train the student model in the original KD process from scratch. **Right**: We transfer the condensed knowledge encoding to facilitate another KD process.

Reuse of Condensed Knowledge. The observed similarity of knowledge value across KD processes inspires us to investigate on reusing the condensed knowledge for efficient training. As shown in Fig. 4(**Left**), we first use the ready-made condensed knowledge to re-distill the student from a scratch. "+DS" (direct selection) and "+VAKS" (value-adaptive knowledge summary) denote two variants of our KCD. It appears that compared with our standard KCD, the performance of the re-distilled student drops significantly when equipped with "+DS" while achieves comparable results with "+VAKS". As shown in Fig. 4(**Right**), we further evaluate the transferability of the knowledge condensation, where we transfer the knowledge encoding condensed in a source KD process to a target KD process to improve the efficiency. As can be seen, the performance of transferring the condensed knowledge degrades dramatically compared with the standard KCD. In comparison, when equipping the transfer process with our VASK module, the performance gap with standard KCD is reduced a lot. These observations demonstrate the potential of our KCD method for promoting the efficient training by reusing and transferring the condensed knowledge.

5 Conclusion

This paper proposes Knowledge Condensation Distillation (KCD) to address the knowledge redundancy during KD. Instead of relying on the whole knowledge set from teacher model, the key idea is to first identify the informative knowledge components and then summarize a compact knowledge encoding to perform KD efficiently. Specially, we forge an iterative optimization framework between condensing the compact knowledge encoding and compressing the student model based on the EM algorithm. We further present two collaborative modules to perform the proposed KCD, as online global value estimation (OGVE) and value-adaptive knowledge summary (VAKS). Extensive experiments demonstrate the effectiveness of the proposed KCD against the state-of-the-arts.

Acknowledgement. The study is supported partly by the National Natural Science Foundation of China under Grants 82172033, U19B2031, 61971369, 52105126, China, in part of Science and Technology Key Project of Fujian Province(No. 2019HZ020009).

References

1. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational information distillation for knowledge transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9163–9171 (2019)
2. Cai, H., Zhu, L., Han, S.: Proxylessnas: Direct neural architecture search on target task and hardware. In: Proceedings of the International Conference of Learning Representation (ICLR) (2019)
3. Chen, L., Wang, D., Gan, Z., Liu, J., Henao, R., Carin, L.: Wasserstein contrastive representation distillation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16296–16305 (2021)
4. Chen, P., Liu, S., Zhao, H., Jia, J.: Distilling knowledge via knowledge review. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5008–5017 (2021)
5. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 248–255 (2009)
6. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
7. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels. Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) **31** (2018)
8. Har-Peled, S., Kushal, A.: Smaller coresets for k-median and k-means clustering. Discrete & Computational Geometry **37**(1), 3–19 (2007)
9. Heo, B., Kim, J., Yun, S., Park, H., Kwak, N., Choi, J.Y.: A comprehensive overhaul of feature distillation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1921–1930 (2019)
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
11. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
12. Katharopoulos, A., Fleuret, F.: Not all samples are created equal: Deep learning with importance sampling. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 2525–2534 (2018)
13. Komodakis, N., Zagoruyko, S.: Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer. In: Proceedings of the International Conference of Learning Representation (ICLR) (2017)
14. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
15. Li, J., Zhou, P., Xiong, C., Hoi, S.C.: Prototypical contrastive learning of unsupervised representations. In: Proceedings of the International Conference of Learning Representation (ICLR
16. Li, S., Lin, M., Wang, Y., Fei, C., Shao, L., Ji, R.: Learning efficient gans for image translation via differentiable masks and co-attention distillation. IEEE Transactions on Multimedia (TMM) (2022)
17. Li, S., Lin, M., Wang, Y., Wu, Y., Tian, Y., Shao, L., Ji, R.: Distilling a powerful student model via online knowledge distillation. IEEE Transactions on Neural Networks and Learning Systems (TNNLS) (2022)

18. Lin, M., Ji, R., Wang, Y., Zhang, Y., Zhang, B., Tian, Y., Shao, L.: Hrank: Filter pruning using high-rank feature map. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1529–1538 (2020)
19. Ma, N., Zhang, X., Zheng, H.T., Sun, J.: Shufflenet v2: Practical guidelines for efficient cnn architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 116–131 (2018)
20. Mirzasoleiman, B., Bilmes, J., Leskovec, J.: Coresets for data-efficient training of machine learning models. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 6950–6960 (2020)
21. Müller, R., Kornblith, S., Hinton, G.E.: When does label smoothing help? Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2019)
22. Olvera-López, J.A., Carrasco-Ochoa, J.A., Martínez-Trinidad, J.F., Kittler, J.: A review of instance selection methods. Artificial Intelligence Review **34**(2), 133–143 (2010)
23. Park, W., Kim, D., Lu, Y., Cho, M.: Relational knowledge distillation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3967–3976 (2019)
24. Passalis, N., Tefas, A.: Learning deep representations with probabilistic knowledge transfer. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 268–284 (2018)
25. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
26. Sener, O., Savarese, S.: Active learning for convolutional neural networks: A coreset approach. arXiv preprint arXiv:1708.00489 (2017)
27. Shen, Z., Liu, Z., Xu, D., Chen, Z., Cheng, K.T., Savvides, M.: Is label smoothing truly incompatible with knowledge distillation: An empirical study. In: Proceedings of the International Conference of Learning Representation (ICLR) (2020)
28. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
29. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
30. Tian, Y., Krishnan, D., Isola, P.: Contrastive representation distillation. In: Proceedings of the International Conference of Learning Representation (ICLR) (2019)
31. Toneva, M., Sordoni, A., des Combes, R.T., Trischler, A., Bengio, Y., Gordon, G.J.: An empirical study of example forgetting during deep neural network learning. In: Proceedings of the International Conference of Learning Representation (ICLR) (2018)
32. Tung, F., Mori, G.: Similarity-preserving knowledge distillation. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). pp. 1365–1374 (2019)
33. Wang, T., Zhu, J.Y., Torralba, A., Efros, A.A.: Dataset distillation. arXiv preprint arXiv:1811.10959 (2018)
34. Xu, G., Liu, Z., Li, X., Loy, C.C.: Knowledge distillation meets self-supervision. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 588–604 (2020)
35. Xu, G., Liu, Z., Loy, C.C.: Computation-efficient knowledge distillation via uncertainty-aware mixup. arXiv preprint arXiv:2012.09413 (2020)

36. Yamamoto, K.: Learnable companding quantization for accurate low-bit neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5029–5038 (2021)
37. Yin, H., Molchanov, P., Alvarez, J.M., Li, Z., Mallya, A., Hoiem, D., Jha, N.K., Kautz, J.: Dreaming to distill: Data-free knowledge transfer via deepinversion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
38. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146 (2016)
39. Zhang, X., Zhou, X., Lin, M., Sun, J.: Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6848–6856 (2018)
40. Zhang, Z., Chen, X., Chen, T., Wang, Z.: Efficient lottery ticket finding: Less data is more. In: Proceedings of the International Conference on Machine Learning (ICML). pp. 12380–12390 (2021)
41. Zhao, B., Mopuri, K.R., Bilen, H.: Dataset condensation with gradient matching. In: Proceedings of the International Conference of Learning Representation (ICLR) (2020)
42. Zhu, X., Gong, S., et al.: Knowledge distillation by on-the-fly native ensemble. Proceedings of the Advances in Neural Information Processing Systems (NeurIPS) (2018)