Distribution Shift Matters for Knowledge Distillation with Webly Collected Images

Jialiang Tang[1,2,3], Shuo Chen[4,*], Gang Niu[4], Masashi Sugiyama[4,5], Chen Gong[1,2,3,*]
[1]School of Computer Science and Engineering, Nanjing University of Science and Technology, China
[2]Key Laboratory of Intelligent Perception and Systems for
High-Dimensional Information of Ministry of Education, China
[3]Jiangsu Key Laboratory of Image and Video Understanding for Social Security, China
[4]Center for Advanced Intelligence Project, RIKEN, Japan
[5]The Graduate School of Frontier Sciences, The University of Tokyo, Japan

Abstract

Knowledge distillation aims to learn a lightweight student network from a pre-trained teacher network. In practice, existing knowledge distillation methods are usually infeasible when the original training data is unavailable due to some privacy issues and data management considerations. Therefore, data-free knowledge distillation approaches proposed to collect training instances from the Internet. However, most of them have ignored the common distribution shift between the instances from original training data and webly collected data, affecting the reliability of the trained student network. To solve this problem, we propose a novel method dubbed "Knowledge Distillation between Different Distributions" (KD^3), which consists of three components. Specifically, we first dynamically select useful training instances from the webly collected data according to the combined predictions of teacher network and student network. Subsequently, we align both the weighted features and classifier parameters of the two networks for knowledge memorization. Meanwhile, we also build a new contrastive learning block called MixDistribution to generate perturbed data with a new distribution for instance alignment, so that the student network can further learn a distribution-invariant representation. Intensive experiments on various benchmark datasets demonstrate that our proposed KD^3 can outperform the state-of-the-art data-free knowledge distillation approaches.

1. Introduction

In recent years, advanced deep neural networks (DNNs) have significantly succeeded in many computer vision fields [19, 21]. However, those excellent DNNs usually have excess learning parameters, which may incur unaffordable computation and memory burdens for resource-limited intelligent devices. To address this problem, model compression algorithms have been developed to constrict heavy DNNs into portable ones, mainly including the network pruning [28], network quantization [35], and knowledge distillation [23].

Most existing compression algorithms are data-driven and rely on massive original training data that is usually inaccessible in the real world. For example, the large-scale ImageNet [12] requires 138*GB* of storage and is too heavy to transfer among devices, yet the ResNet34 [22] trained on ImageNet only needs 85*MB* memory and can be shared at a relatively low cost. Besides, users may be more willing to share pre-trained models than their personal data, such as photos and travel records. As a result, existing data-driven algorithms for model compression frequently fail to deal with large DNNs in practical applications.

To address this issue, data-free model compression methods have received wide attention in recent studies [6, 10, 13, 16]. Among these methods, data-free knowledge distillation has shown encouraging results, which only requires a pre-trained large network (*a.k.a.* a teacher network) to learn a compact network (*a.k.a.* a student network). Existing data-free knowledge distillation methods train student network with the guidance of teacher network through the generated pseudo data [7, 45, 50] or real-world data collected from the Internet [6]. Generally, the performance of the student networks trained on synthetic data might be suboptimal due to the flawed or distorted synthetic images. In comparison, the student networks using real-world data from the Internet usually achieve better performance, especially on the tasks involving complicated natural images.

Current data-free knowledge distillation methods [6] that

[*]Corresponding authors: Chen Gong (chen.gong@njust.edu.cn), Shuo Chen (shuo.chen.ya@riken.jp).

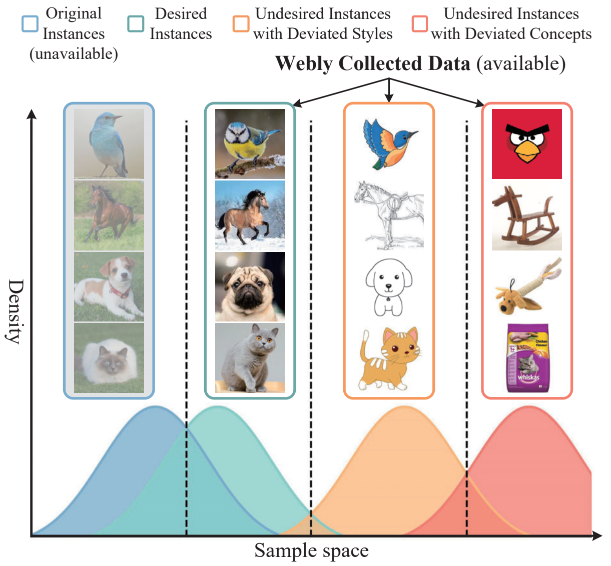

Original Instances (unavailable)
Desired Instances
Undesired Instances with Deviated Styles
Undesired Instances with Deviated Concepts
Webly Collected Data (available)
Density
Sample space

Figure 1. The illustration of distribution shift between the webly collected data and original data, where the original data consists of realistic images of animals. Nevertheless, the webly collected data may include cartoon and sketch images of animals, and even some non-animal images.

train student network with data from the Internet (*i.e.*, webly collected data) seek to select confident instances from the collected data, so that they can provide correctly labeled images for training student network. However, the webly collected data and original data may have different distributions, and existing methods usually ignore the distribution shift (*e.g.*, the image style and image category) between them, as shown in Fig. 1. For example, when we are interested in classifying various real-world animals and enter “cat” into the image search engines, we may obtain the images of “cartoon cat” or “cat food”. Apparently, the former is with different styles of cat images, and the latter is even unrelated to our interested animal classification task. The student network trained on the webly collected data will inevitably suffer from distribution shift when it is evaluated on the unseen test data. This makes the performance of student network trained on the webly collected data obviously lower than that using the original data. Consequently, it is critically important to alleviate the distribution shift between the webly collected data and original data.

To this end, we propose a new data-free approach called **K**nowledge **D**istillation between **D**ifferent **D**istributions (KD^3) to learn a student network by utilizing the plentiful data collected from the Internet with specific considerations on the distribution shift. More specifically, we first select the webly collected instances with the similar distribution to original data by dynamically combining the predictions of teacher network and student network during the training phase. After that, to exhaustively learn the information of teacher network, we share the classifier of teacher network with student network and conduct a weighted feature alignment. In this way, we can encourage student network to mimic the feature extraction of teacher network. Furthermore, a new contrastive learning block MixDistribution is designed to control the statistics (*i.e.*, the mean and variance) of instances, so that we can generate perturbed instances with the new distribution. The student network is encouraged to produce consistent features for the unperturbed and perturbed instances to learn the distribution-invariant representation, which can generalize to the previously unseen test data. As a result, the student network that precisely mimics teacher network can produce the features that are consistent with teacher network. Finally, these features fed into the shared classifier can make the predictions as accurate as the corresponding teacher network. Thanks to effectively resolving the distribution shift between the webly collected data and original data, our KD^3 finally learns an accurate and lightweight student network, which can achieve comparable performance to those student networks trained on the original data. The contributions of our proposed KD^3 are summarized as follows:

- We propose a new data-free knowledge distillation method termed KD^3, which dynamically selects useful training instances from the Internet by alleviating the distribution shift between the original data and webly collected data.
- We design a weighted feature alignment strategy and a new contrastive learning block to closely match student network with teacher network in the feature space, so that the student network can successfully learn useful knowledge from teacher network for the unseen original data.
- Intensive experiments on multiple benchmarks demonstrate that our KD^3 can outperform the state-of-the-art data-free knowledge distillation approaches.

2. Related Works

In this section, we review previous works related to our proposed KD^3, mainly including knowledge distillation and the learning approaches under distribution shift.

2.1. Knowledge Distillation

Conventional knowledge distillation usually needs the original training data to launch knowledge transfer from a teacher to a student. In general, they utilize the soften predictions [1, 23], middle-layer features [5, 36], and instance relationships [33, 40] as the transferred knowledge, which can achieve satisfactory results on various datasets and different DNNs. However, they are usually ineffective in practice when the original data is unusable.

To solve this problem, data-free knowledge distillation [4, 48] employs synthetic data or webly collected data to train student network with the help of the pre-trained teacher network, which can bypass privacy issues and save data management costs in practical applications. Inspired by the Generative Adversarial Networks [20], a series of works [7, 15, 30] treat the teacher network as the discriminator to supervise a generator to produce pseudo data from random noise. Besides, DeepInversion [45] extracts the means and variances stored in the batch normalization layers of teacher network to reconstruct training images. Recently, Contrastive Model Inversion (CMI) [16] argues that the instances generated by DeepInversion are highly similar, which is ineffective for student network training. Consequently, CMI augments the diversity of generated data via contrastive learning [9]. Lately, Zhao et al. [50] use the means and variances of teacher network to guide the generator and further produce new realistic data, thereby improving the performance of student network.

Instead of generating new data for approximating the original data, it is promising to train a satisfactory student network by utilizing the plentiful realistic instances on the Internet. Xu et al. [44] select useful examples from the webly collected data based on a portion of the original data. Chen et al. [6] propose to select useful instances with a low cross-entropy value to train student network. However, they neglect the distribution discrepancies between the webly collected data and original data, which inevitably corrupts the performance of the student network. In this work, we carefully consider and effectively process the distribution shift, thus obtaining a reliable student network.

2.2. Learning under Distribution Shift

In the learning scenarios with distribution shift, the training data and test data may come from different distributions [32, 34]. In this case, DNNs are biased to training data and cannot perform well during the test phase. To tackle this problem, a series of works [2, 42] propose to select instances that are similar to the target distribution, and those selected instances are used for retraining the DNNs. Some other works [17, 39] adapt the reweighting technique to find out the useful training instances which have the similar distribution to the original data. Furthermore, domain adaption approaches [14, 49] are proposed to transfer knowledge from the training data (*i.e.*, the source data) to test data (*i.e.*, the target data) , thereby improving the generalization ability of the model under distribution shift.

In data-free knowledge distillation, the distribution of webly collected data is usually different from the unseen test data, which may drop the performance of student network significantly. Therefore, we propose the new method KD^3 to explicitly deal with such a distribution shift issue for data-free knowledge distillation.

3. Our Approach

In this section, we first introduce some necessary preliminary knowledge, and then we state our KD^3 on how to learn student networks without using the original data.

3.1. Preliminary

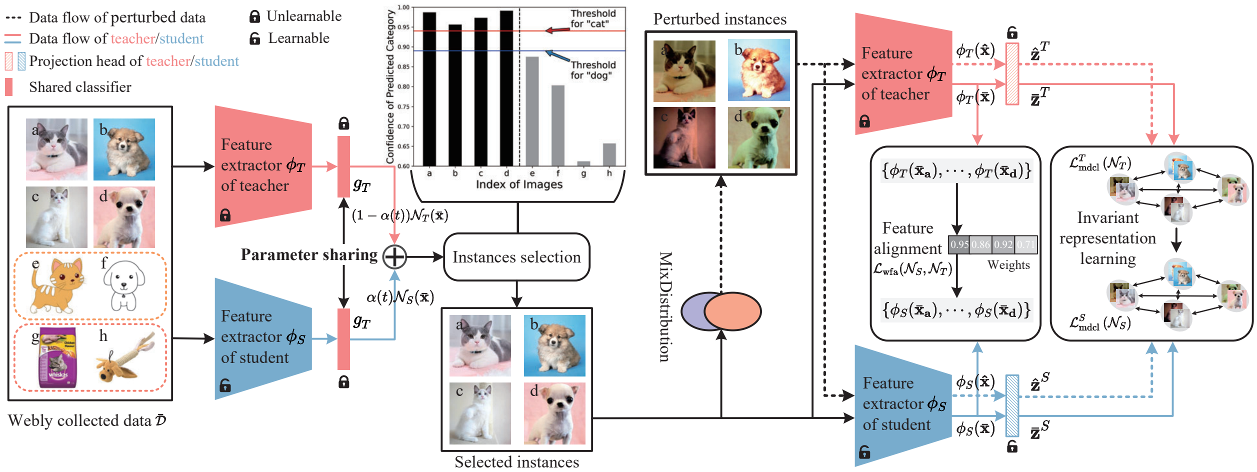
Conventional knowledge distillation methods [23, 36] seek to learn a small student network $\mathcal{N}_S$ by promoting it to mimic the output of a large pre-trained teacher network $\mathcal{N}_T$. Formally, we denote the original training data as $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|\mathcal{D}|} \subset \mathcal{X} \times \mathcal{Y}$, where "$|\cdot|$" is the data cardinality; $\mathcal{X} \subset \mathbb{R}^I$ (I is the data dimensionality) and $\mathcal{Y} = \{1, \cdots, K\}$ (K is the total number of classes) are the sample space and label space, respectively. For a training dataset $\mathcal{D}$, the knowledge distillation is accomplished by minimizing the following loss function:

$$\mathcal{L}_{\text{kd}}(\mathcal{N}_S) = \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} [\mathcal{H}_{\text{ce}}\left(\mathcal{N}_S\left(\mathbf{x}_i\right), y_i\right) + \lambda \mathcal{H}_{\text{kt}}\left(\mathbf{f}_i^S, \mathbf{f}_i^T\right)], \tag{1}$$

where $\mathcal{H}_{\text{ce}}$ is the cross-entropy loss function, encouraging the prediction of student network to be as consistent as the ground-truth; $\mathcal{H}_{\text{kt}}$ is the knowledge transfer function to promote student network to learn the knowledge $\mathbf{f}_i^T$ of teacher network (*e.g.*, predictions or feature maps); $\mathbf{f}_i^S$ is the corresponding knowledge of student network; $\lambda > 0$ denotes the trade-off parameter, which is used to balance $\mathcal{H}_{\text{ce}}$ and $\mathcal{H}_{\text{kt}}$.

The necessary original data $\mathcal{D}$ of conventional knowledge distillation methods is usually untouchable due to practical limitations discussed in Section 1. Consequently, a sequence of data-free methods [6, 7, 10] propose to generate the pseudo data from teacher network $\mathcal{N}_T$, but the visual quality and diversity of the synthetic images limit their performance. Instead of generating pseudo data, there are massive realistic data $\bar{\mathcal{D}} = \{(\bar{\mathbf{x}}_i, \bar{y}_i)\}_{i=1}^{|\bar{\mathcal{D}}|} \subset \bar{\mathcal{X}} \times \bar{\mathcal{Y}}$ on the Internet which can be gathered to train the student network $\mathcal{N}_S$ [8]. Here, the notations with superscript "–" denote that they are related to the webly collected data. However, there is distribution shift between the webly collected data $\bar{\mathcal{D}}$ and original data $\mathcal{D}$, namely: 1) $p(y|\mathbf{x}) \neq p(\bar{y}|\bar{\mathbf{x}})$, *i.e.*, $\bar{\mathcal{D}}$ may contain many uninterested instances due to $\mathcal{Y} \subset \bar{\mathcal{Y}}$ and $|\mathcal{D}| \ll |\bar{\mathcal{D}}|$; 2) $p(\mathbf{x}) \neq p(\bar{\mathbf{x}})$, *i.e.*, the image quality or style of $\mathcal{D}$ and $\bar{\mathcal{D}}$ are different from each other because the instances in $\bar{\mathcal{D}}$ are roughly collected from the Internet. In this case, the student network trained on $\bar{\mathcal{D}}$ inevitably performs poorly on the unseen test data due to the distribution shift.

To address the aforementioned issue, we propose a novel data-free knowledge distillation method called KD^3 to train a reliable $\mathcal{N}_S$ on the webly collected data $\bar{\mathcal{D}}$. As illustrated in Fig. 2, our KD^3 contains three key components (as detailed in Sections 3.2, 3.3, and 3.4, respectively), includ-



Data flow of perturbed data
Data flow of teacher/student
Projection head of teacher/student
Shared classifier
Unlearnable
Learnable
a
b
c
d
e
f
g
h
Webly collected data $\bar{\mathcal{D}}$
Feature extractor ϕ_T of teacher
Feature extractor ϕ_S of student
g_T
g_T
Parameter sharing
$(1-\alpha(t))\mathcal{N}_T(\bar{\mathbf{x}})$
$\alpha(t)\mathcal{N}_S(\bar{\mathbf{x}})$
1.00
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
Confidence of Predicted Category
Threshold for "cat"
Threshold for "dog"
Index of Images
Instances selection
Selected instances
Perturbed instances
MixDistribution
Feature extractor ϕ_T of teacher
$\phi_T(\hat{\mathbf{x}})$
$\phi_T(\bar{\mathbf{x}})$
$\hat{\mathbf{z}}^T$
$\bar{\mathbf{z}}^T$
$\{\phi_T(\bar{\mathbf{x}}_\mathbf{a}),\cdots,\phi_T(\bar{\mathbf{x}}_\mathbf{d})\}$
Feature alignment
$\mathcal{L}_{\text{wfa}}(\mathcal{N}_S,\mathcal{N}_T)$
0.95
0.86
0.92
0.71
Weights
$\{\phi_S(\bar{\mathbf{x}}_\mathbf{a}),\cdots,\phi_S(\bar{\mathbf{x}}_\mathbf{d})\}$
$\mathcal{L}^T_{\text{mdcl}}(\mathcal{N}_T)$
Invariant representation learning
$\mathcal{L}^S_{\text{mdcl}}(\mathcal{N}_S)$
Feature extractor ϕ_S of student
$\phi_S(\hat{\mathbf{x}})$
$\phi_S(\bar{\mathbf{x}})$
$\hat{\mathbf{z}}^S$
$\bar{\mathbf{z}}^S$

Figure 2. The diagram of our **K**nowledge **D**istillation between **D**ifferent **D**istributions (KD^3). The frozen teacher network $\mathcal{N}_T$ consists of a feature extractor ϕ_T and a classifier g_T. The student network $\mathcal{N}_S$ has a learnable feature extractor ϕ_S and shares g_T with $\mathcal{N}_T$, where g_T is fixed to preserve the information learned by $\mathcal{N}_T$. Firstly, the webly collected data $\bar{\mathcal{D}}$ is dynamically selected by $\mathcal{N}_T$ and $\mathcal{N}_S$. For example, here we assume the original data contain two classes including: "cat" and "dog". Then we select the images "a-d" while discarding the images "e-h" (with different distributions). Subsequently, the weighted feature alignment conducted on the selected data further promotes $\mathcal{N}_S$ to make accurate predictions. Moreover, the MixDistribution contrastive learning is applied to both the perturbed instances (obtained by the MixDistribution) and unperturbed instances, promoting $\mathcal{N}_S$ to learn robust representations that are invariant to distribution shift.

ing 1) Teacher-student dynamic instance selection, which chooses webly collected instances having the similar distribution of original instances; 2) Classifier sharing & feature alignment, where the student network and teacher network share their classifier parameters and align their output features; 3) MixDistribution contrastive learning, which promotes the student network to produce consistent representations for both perturbed and unperturbed instances.

3.2. Teacher-Student Dynamic Instance Selection

As mentioned above, there is distribution shift between the webly collected data $\bar{\mathcal{D}}=\{(\bar{\mathbf{x}}_i,\bar{y}_i)\}_{i=1}^{|\bar{\mathcal{D}}|}$ and original data $\mathcal{D}$. Since the teacher network is well-trained on $\mathcal{D}$, it is able to show high confidence levels for those instances in $\bar{\mathcal{D}}$ which have the similar distribution with $\mathcal{D}$. Consequently, we propose to select useful instances from $\bar{\mathcal{D}}$ based on the output probabilities of teacher network $\mathcal{N}_T$ and student network $\mathcal{N}_S$ to alleviate the distribution shift.

Specifically, we first input all instances in $\bar{\mathcal{D}}$ into both $\mathcal{N}_T$ and $\mathcal{N}_S$ to get the corresponding output probabilities:

$$\begin{cases} \mathcal{N}_T(\bar{\mathbf{x}}_i) = g_T(\phi_T(\bar{\mathbf{x}}_i)), \\ \mathcal{N}_S(\bar{\mathbf{x}}_i) = g_T(\phi_S(\bar{\mathbf{x}}_i)), \end{cases} \tag{2}$$

where ϕ_T and ϕ_S denote the feature extractor of $\mathcal{N}_T$ and $\mathcal{N}_S$, respectively, and g_T represents the shared classifier learned by $\mathcal{N}_T$. Then, we combine the predictions $\{\mathcal{N}_T(\bar{\mathbf{x}}_i)\}_{i=1}^{|\bar{\mathcal{D}}|}$ and $\{\mathcal{N}_S(\bar{\mathbf{x}}_i)\}_{i=1}^{|\bar{\mathcal{D}}|}$ by the following criterion:

$$\text{Combine}\left(\bar{\mathbf{x}}_i\right) = \left(1-\alpha(t)\right)\mathcal{N}_T\left(\bar{\mathbf{x}}_i\right) + \alpha(t)\mathcal{N}_S\left(\bar{\mathbf{x}}_i\right). \tag{3}$$

The combination is dynamically adjusted by the following time-dependent function:

$$\alpha(t) = \begin{cases} \exp\left(-5\left(\frac{t}{I/2}-1\right)^2\right), & t \le I/2, \\ 1, & t > I/2, \end{cases} \tag{4}$$

where $\alpha(t)$ grows from 0 to 1 according to current epoch t, and I represents the total number of iterations in training student network. In the early-staged training, the initialized $\mathcal{N}_S$ is unable to offer accurate predictions for the instances in $\bar{\mathcal{D}}$, while the pre-trained $\mathcal{N}_T$ can precisely recognize the instances in $\bar{\mathcal{D}}$ which are similar to $\mathcal{D}$. Therefore, function $\alpha(t)$ attributes a big weight to $\mathcal{N}_T\left(\bar{\mathbf{x}}_i\right)$ at the early stage. With the improvement of $\mathcal{N}_S$, $\alpha(t)$ will gradually highlight the importance of $\mathcal{N}_S\left(\bar{\mathbf{x}}_i\right)$. When $\alpha(t)=1$ (namely $t \ge I/2$), the selection of training instances will be completely determined by the student network $\mathcal{N}_S$.

Subsequently, we can obtain the predicted label y_i^{pred} of image $\bar{\mathbf{x}}_i$ and the confidence p_i of $\bar{\mathbf{x}}_i$ belong to y_i^{pred} as:

$$\begin{aligned} y_i^{\text{pred}} &= \arg\max_j \left(\text{Combine}\left(\bar{\mathbf{x}}_i\right)\right)_j, \\ p_i &= \left(\text{Combine}\left(\bar{\mathbf{x}}_i\right)\right)_{y_i^{\text{pred}}}, \end{aligned} \tag{5}$$

respectively. Based on the above $\{y_i^{\text{pred}}\}_{i=1}^{|\bar{\mathcal{D}}|}$, we can count the number of labels for each class as $\{n_i\}_{i=1}^{K}$, and then we obtain the thresholds $\{\mathcal{T}_i\}_{i=1}^{K}$ for filtering out the low-confidence instances of each category as:

$$\mathcal{T}_i = \text{Normalization}(n_i)\cdot V_{\text{th}}, \tag{6}$$

where V_{th} is a fixed threshold, and $\{n_i\}_{i=1}^{K}$ is normalized to $[0, 1]$ via the following rule:

$$\text{Normalization}(n_i) = \frac{n_i}{\max_{1\leq j\leq K}(n_j)}. \tag{7}$$

Finally, we obtain the useful data $\bar{\mathcal{D}}_s = \text{Select}(\bar{\mathbf{x}}_i)$ $(1 \leq i \leq |\bar{\mathcal{D}}|)$ which has the similar distribution with $\mathcal{D}$. Here, the selection operator is defined as:

$$\text{Select}(\bar{\mathbf{x}}_i) = \begin{cases} \bar{\mathbf{x}}_i \in \bar{\mathcal{D}}_s, & p_i > \mathcal{T}_{y_i^{\text{pred}}}, \\ \bar{\mathbf{x}}_i \notin \bar{\mathcal{D}}_s, & p_i \leq \mathcal{T}_{y_i^{\text{pred}}}. \end{cases} \tag{8}$$

During the training phase, $\bar{\mathcal{D}}_s$ is continuously updated based on $\mathcal{N}_T$, $\mathcal{N}_S$, and $\alpha(t)$. If $\mathcal{N}_S$ performs worse on a certain category, it will produce low confidence values for the instances of this category and lead to a low threshold. In this case, many instances belonging to this category can be selected to supplement the training of $\mathcal{N}_S$ for this category.

3.3. Classifier Sharing & Feature Alignment

In Section 3.2, we successfully select the useful data $\bar{\mathcal{D}}_s = \{(\bar{\mathbf{x}}_i, \bar{y}_i)\}_{i=1}^{|\bar{\mathcal{D}}_s|}$ from the webly collected data $\bar{\mathcal{D}}$. However, teacher network $\mathcal{N}_T$ and student network $\mathcal{N}_S$ are unable to make completely correct predictions for all instances in $\bar{\mathcal{D}}$ because both two networks are imperfect, especially in the early-staged iterations. Therefore, the distribution $p(\bar{y}|\bar{\mathbf{x}})$ of $\bar{\mathcal{D}}_s$ is still different from $p(y|\mathbf{x})$ of $\mathcal{D}$ in some cases, which may incur inaccurate supervisions to hurt the performance of student network. Recent works [14, 27] revealed that the classifiers of DNNs can learn task-specific information. Inspired by this, we share the classifier g_T (learned by $\mathcal{N}_T$) with $\mathcal{N}_S$, so that the critical information of unseen original data (contained in $\mathcal{D}$) can be transferred from $\mathcal{N}_T$ to $\mathcal{N}_S$. Furthermore, we freeze g_T to prevent the information learned from the original data being disturbed by parameter update, which means that $\mathcal{N}_S$ only updates its parameters in ϕ_S during training. Subsequently, we utilize $\bar{\mathcal{D}}_s$ to drive feature alignment between $\mathcal{N}_S$ and $\mathcal{N}_T$ in the preceding layer of the shared g_T, so that the student network can memorize critical knowledge of teacher network as much as possible.

In detail, the overall goal of our feature alignment is to encourage $\mathcal{N}_S$ to produce outputs as consistent as that of $\mathcal{N}_T$. Accordingly, we estimate the feature alignment weight w_i for $\bar{\mathbf{x}}_i \in \bar{\mathcal{D}}_s$ by calculating the consistency between $\mathcal{N}_T(\bar{\mathbf{x}}_i)$ and $\mathcal{N}_S(\bar{\mathbf{x}}_i)$, which is:

$$w_i = 1 - \text{Sigmoid}\left(\left\|\mathcal{N}_S(\bar{\mathbf{x}}_i) - \mathcal{N}_T(\bar{\mathbf{x}}_i)\right\|_1\right). \tag{9}$$

Algorithm 1 **K**nowledge **D**istillation between **D**ifferent **D**istributions.

```
Require: A large pre-trained teacher network $\mathcal{N}_T$, webly collected data $\bar{\mathcal{D}} = \{(\bar{\mathbf{x}}_i, \bar{y}_i)\}_{i=1}^{|\bar{\mathcal{D}}|}$, trade-off parameter $\alpha$.
Initialize the small student network $\mathcal{N}_S$;
Share the classifier $g_T$ of $\mathcal{N}_T$ with $\mathcal{N}_S$;
repeat
    Stage 1: Similar distribution data selection.
    Calculate $\alpha(t)$ via Eq. (4);
    Combine outputs of $\mathcal{N}_T$ and $\mathcal{N}_S$ via Eq. (3);
    Calculate thresholds $\{\mathcal{T}_i\}_{i=1}^{K}$ via Eq. (6);
    Sample data $\bar{\mathcal{D}}_s$ from $\bar{\mathcal{D}}$ via Eq. (8);
    Stage 2: Learning student network $\mathcal{N}_S$.
    Measure instance weights $\{w_i\}_{i=1}^{|\bar{\mathcal{D}}_s|}$ via Eq. (9);
    Calculate feature alignment loss $\mathcal{L}_{\text{wfa}}(\mathcal{N}_S, \mathcal{N}_T)$ via Eq. (10);
    Perturbe instances in $\bar{\mathcal{D}}_s$ via Eq. (12);
    Calculate MixDistribution contrastive learning loss $\mathcal{L}_{\text{mdcl}}(\mathcal{N}_S, \mathcal{N}_T)$ via Eq. (17);
    Calculate complete objective loss $\mathcal{L}_{\text{objective}}(\mathcal{N}_S, \mathcal{N}_T)$ via Eq. (18);
    Update student network $\mathcal{N}_S$ via SGD;
until convergence
Ensure: Lightweight student network $\mathcal{N}_S$.
```

For an image $\bar{\mathbf{x}}_i$, if $\mathcal{N}_S$ produces consistent outputs with that of $\mathcal{N}_T$, we regard it as an easily-aligned instance and give it a large weight to highlight its positive influence in feature alignment and vice versa. Based on $\{w_i\}_{i=1}^{|\bar{\mathcal{D}}_s|}$, the **w**eighted **f**eature **a**lignment loss $\mathcal{L}_{\text{wfa}}(\mathcal{N}_S, \mathcal{N}_T)$ is formulated as:

$$\mathcal{L}_{\text{wfa}}(\mathcal{N}_S, \mathcal{N}_T) = \frac{1}{|\bar{\mathcal{D}}_s|}\sum_{i=1}^{|\bar{\mathcal{D}}_s|} w_i \mathcal{H}_{\text{mse}}(\phi_S(\bar{\mathbf{x}}_i), \phi_T(\bar{\mathbf{x}}_i)), \tag{10}$$

where $\mathcal{H}_{\text{mse}}$ is the mean square error and it measures the similarity between $\phi_T(\bar{\mathbf{x}}_i)$ and $\phi_S(\bar{\mathbf{x}}_i)$.

Classifier sharing and feature alignment successfully address the shortage of supervision in the student network, thereby eliminating the negative impact of inaccurate labels caused by the webly collected data. When evaluated on the test instance $\mathbf{x}_{\text{test}}$, the student network well aligned with the teacher network can produce feature $\phi_S(\mathbf{x}_{\text{test}})$ which is consistent to $\phi_T(\mathbf{x}_{\text{test}})$. After that, the parameter-shared classifier g_T can produce an accurate prediction $g_T(\phi_S(\mathbf{x}_{\text{test}}))$ like the teacher prediction $g_T(\phi_T(\mathbf{x}_{\text{test}}))$.

3.4. MixDistribution Contrastive Learning

In our problem setting, the original training data $\mathcal{D}$ of teacher network $\mathcal{N}_T$ is inaccessible and student network $\mathcal{N}_S$ trained on the selected data $\bar{\mathcal{D}}_s$ needs to correctly recognize the unseen test data. In practice, the original data $\mathcal{D}$ is usually selected and processed manually, so the distribution $p(\bar{\mathbf{x}})$ of webly collected instances cannot accurately match the distribution $p(\mathbf{x})$ of the original data. Recent studies [45, 51] find that the data distribution is closely related to image style and quality, which can be reflected in statistical variables, *e.g.*, the standard deviation and mean. Therefore, we propose MixDistribution to construct the perturbed data with new distribution, which disturbs statistics of images in $\bar{\mathcal{D}}_s$. Finally, we promote student network to learn representation that is invariant to distribution shift by improving the consistency between perturbed and unperturbed instances.

More specifically, we first randomize $\{\bar{\mathbf{x}}_i\}_{i=1}^{|\bar{\mathcal{D}}_s|}$ as Randomize $(\{\bar{\mathbf{x}}_i\}_{i=1}^{|\bar{\mathcal{D}}_s|})$ and compute the perturbed statistics by the following rules:

$$\begin{cases} \gamma_{\text{mix}} = \lambda\sigma(\{\bar{\mathbf{x}}_i\}_{i=1}^{|\bar{\mathcal{D}}_s|}) + (1-\lambda)\sigma(\text{Rand}(\{\bar{\mathbf{x}}_i\}_{i=1}^{|\bar{\mathcal{D}}_s|})), \\ \beta_{\text{mix}} = \lambda\mu(\{\bar{\mathbf{x}}_i\}_{i=1}^{|\bar{\mathcal{D}}_s|}) + (1-\lambda)\mu(\text{Rand}(\{\bar{\mathbf{x}}_i\}_{i=1}^{|\bar{\mathcal{D}}_s|})), \end{cases} \tag{11}$$

where $\lambda > 0$ is produced by beta distribution $\text{Beta}(\delta, \delta)$ with $\delta \in (0, \infty)$ being a hyper-parameter. Here, $\sigma(\cdot)$ and $\mu(\cdot)$ denote the standard deviation and mean of the corresponding variables, respectively. Then, we construct the perturbed image $\hat{\mathbf{x}}_i$ by:

$$\hat{\mathbf{x}}_i = \gamma_{\text{mix}} \frac{\bar{\mathbf{x}}_i - \mu(\bar{\mathbf{x}}_i)}{\sigma(\bar{\mathbf{x}}_i)} + \beta_{\text{mix}}, \tag{12}$$

where we scale and shift the normalized $\bar{\mathbf{x}}_i$ by γ_{mix} and β_{mix}, respectively. After the above instance perturbation, the raw images $\{\bar{\mathbf{x}}_i\}_{i=1}^{|\bar{\mathcal{D}}_s|}$ and the perturbed images $\{\hat{\mathbf{x}}_i\}_{i=1}^{|\bar{\mathcal{D}}_s|}$ are fed into $\mathcal{N}_T$ and $\mathcal{N}_S$ to obtain the features in penultimate layer. Subsequently, we follow [9] to transfer features of all dimensionalities into the embedding space by a projection head. By taking the teacher network $\mathcal{N}_T$ and the corresponding feature $\phi_T(\bar{\mathbf{x}}_i)$ as an example, the embedding result $\bar{\mathbf{z}}_i^T$ is calculated by:

$$\bar{\mathbf{z}}_i^T = \text{Normalization}\left(\mathbf{W}_p^T \phi_T(\bar{\mathbf{x}}_i) + \mathbf{b}_p^T\right), \tag{13}$$

where $\mathbf{W}_p^T$ and $\mathbf{b}_p^T$ denote the weight and bias of projection head, and the notations with superscripts "T" and "S" represent they are related to teacher network and student network, respectively. Similarly, the embedding result $\hat{\mathbf{z}}_i^T$ of perturbed example $\hat{\mathbf{x}}_i$ is computed by:

$$\hat{\mathbf{z}}_i^T = \text{Normalization}\left(\mathbf{W}_p^T \phi_T(\hat{\mathbf{x}}_i) + \mathbf{b}_p^T\right). \tag{14}$$

Based on the embeddings of unperturbed instances $\{\bar{\mathbf{z}}_i^T\}_{i=1}^{|\bar{\mathcal{D}}_s|}$ and perturbed instances $\{\hat{\mathbf{z}}_i^T\}_{i=1}^{|\bar{\mathcal{D}}_s|}$, we can calculate the **M**ix**D**istribution **c**ontrastive **l**earning (MDCL) loss $\mathcal{L}_{\text{mdcl}}^T(\mathcal{N}_T)$ for the teacher network as follows:

$$\mathcal{L}_{\text{mdcl}}^T(\mathcal{N}_T) = -\sum_{i=1}^{|\bar{\mathcal{D}}_s|} \log \frac{\exp\left(\text{sim}\left(\bar{\mathbf{z}}_i^T, \hat{\mathbf{z}}_i^T\right)/\tau\right)}{\sum_{j=1}^{|\bar{\mathcal{D}}_s|} \mathbb{1}_{[j \neq i]} \exp\left(\text{sim}\left(\bar{\mathbf{z}}_i^T, \hat{\mathbf{z}}_j^T\right)/\tau\right)}, \tag{15}$$

where $\tau > 0$ is a temperature parameter; $\text{sim}(\cdot)$ denotes the well-known cosine similarity [43]; $\mathbb{1}_{[j \neq i]}$ is the indicator function, and its value is 0 only if $i = j$, and its value is 1, otherwise. Likewise, the MDCL loss of student network is:

$$\mathcal{L}_{\text{mdcl}}^S(\mathcal{N}_S) = -\sum_{i=1}^{|\bar{\mathcal{D}}_s|} \log \frac{\exp\left(\text{sim}\left(\bar{\mathbf{z}}_i^S, \hat{\mathbf{z}}_i^S\right)/\tau\right)}{\sum_{j=1}^{|\bar{\mathcal{D}}_s|} \mathbb{1}_{[j \neq i]} \exp\left(\text{sim}\left(\bar{\mathbf{z}}_i^S, \hat{\mathbf{z}}_j^S\right)/\tau\right)}. \tag{16}$$

$\mathcal{L}_{\text{mdcl}}^T(\mathcal{N}_T)$ and $\mathcal{L}_{\text{mdcl}}^S(\mathcal{N}_S)$ depict the relationship among the embeddings of $\mathcal{N}_T$ and $\mathcal{N}_S$, respectively. Recent studies [33, 40, 52] have demonstrated that transferring the relationship between representations is more effective than transferring representations directly. Therefore, by following [52], we integrate the following learning objectives of $\mathcal{N}_T$ and $\mathcal{N}_S$ based on the similarity relationship:

$$\mathcal{L}_{\text{mdcl}}(\mathcal{N}_S, \mathcal{N}_T) = \mathcal{L}_{\text{mdcl}}^S(\mathcal{N}_S) + \mathcal{L}_{\text{mdcl}}^T(\mathcal{N}_T). \tag{17}$$

By minimizing $\mathcal{L}_{\text{mdcl}}(\mathcal{N}_S, \mathcal{N}_T)$, the student network is encouraged to produce close representations for perturbed and unperturbed versions of the same instance, despite distribution shift between the two versions. Therefore, the student network can accurately classify the test instances of which the distribution is different from $\bar{\mathcal{D}}_s$. Note that the parameters in the projection heads of teacher and student will keep updating during the training phase.

Overall Learning Objective. In the end, the complete objective function of our KD^3 is:

$$\mathcal{L}_{\text{objective}}(\mathcal{N}_S, \mathcal{N}_T) = \mathcal{L}_{\text{wfa}}(\mathcal{N}_S, \mathcal{N}_T) + \alpha\mathcal{L}_{\text{mdcl}}(\mathcal{N}_S, \mathcal{N}_T), \tag{18}$$

where α is a non-negative trade-off parameter to balance the weighted feature alignment loss $\mathcal{L}_{\text{wfa}}(\mathcal{N}_S, \mathcal{N}_T)$ and MixDistribution contrastive learning loss $\mathcal{L}_{\text{mdcl}}(\mathcal{N}_S, \mathcal{N}_T)$.

The detailed training algorithm of KD^3 is summarized in Alg. 1, which includes two main stages. In the first stage, the useful instances with the similar distribution of original training data are selected from the webly collected data. In the second stage, the features of the student network (shares the classifier of teacher network) are closely matched with that of teacher network, and the MixDistribution contrastive learning is deployed on the student network to learn a representation that is invariant to distribution shift.

4. Experiments

In this section, we demonstrate the effectiveness of our proposed KD^3 on multiple image classification datasets.

Compared Methods: We compare our proposed KD^3 with representative data-free methods, including Data-Free Learning (DAFL) [7], Data-Free Adversarial Learning (DFAD) [15], Dual Discriminator Adversarial Distillation (DDAD) [50], DeepInversion (DI) [45], Zero-Shot

Dataset	Arch	#params^T	#params^S	FLOPs^T	FLOPs^S	ACC^T	ACC^S	DAFL	DFAD	DDAD	DI	ZSKT	PRE	DFQ	CMI	DFND	KD^3	ACC↑
MNIST	∇	0.062M	0.019M	0.42M	0.14M	98.91	98.65	98.20	98.31	98.09	–	97.44	98.33	97.49	–	98.37	**98.76**	+0.39
CIFAR10	♢	21.28M	11.17M	1.16G	0.56G	95.70	95.20	92.22	93.30	93.08	93.26	93.32	93.25	94.61	94.84	94.02	**95.21**	+0.37
	♡	14.73M	9.42M	0.40G	0.28G	94.07	92.69	86.92	90.38	90.85	85.27	91.22	91.82	91.36	88.49	92.61	**94.13**	+1.52
CIFAR100	♢	21.28M	11.17M	1.16G	0.56G	78.05	77.10	74.47	67.70	73.64	61.32	67.74	74.19	77.01	77.04	76.35	**78.44**	+1.40
	♡	14.73M	9.42M	0.40G	0.28G	74.53	72.28	65.36	64.90	68.33	60.00	58.33	70.34	62.53	59.70	70.88	**74.21**	+3.33
CINIC	♢	21.28M	11.17M	1.16G	0.56G	86.62	85.09	60.54	71.38	80.10	78.57	64.73	77.56	71.76	78.47	82.96	**86.55**	+3.59
	♡	14.73M	9.42M	0.40G	0.28G	84.22	83.28	59.08	60.67	77.90	68.90	58.84	65.38	74.33	74.99	81.82	**83.54**	+1.72
TinyImageNet	♢	21.28M	11.17M	4.65G	2.23G	66.44	64.87	52.20	20.63	59.84	6.98	31.51	50.15	63.73	64.01	60.92	**66.24**	+2.23
	♡	14.73M	9.42M	1.26G	0.92G	62.34	61.55	53.89	38.95	42.25	1.22	30.63	45.92	23.43	17.73	56.87	**61.98**	+5.11

Table 1. Classification accuracy (in %) of the student network trained by various methods on five image classification datasets. The notations ∇, ♢, and ♡ represent the teacher-student pairs LeNet5-LeNet5_half, ResNet34-ResNet18, and VGGNet16-VGGNet13, respectively. ACC, #params, and floating point operations (FLOPs) denote the yielded accuracy, parameters (in millions, M), and calculations (in Gigas, G) of the corresponding DNN, respectively. These notations with superscripts "T" and "S" represent that they are related to the teacher network and student network, respectively. The best results achieved by baseline methods are underlined, and the column "ACC↑" with green fonts shows the accuracy improvement of KD^3 in contrast to the best results among compared baseline methods.

Knowledge Transfer (ZSKT) [30], Pseudo Replay Enhanced Data-Free Knowledge Distillation (PRE) [3], Data-Free Quantization (DFQ) [10], Contrastive Model Inversion (CMI) [16], and Data-Free Noisy Distillation (DFND) [6] (the only existing method using the webly collected data). We implement the above methods by using the codes on their official GitHub pages.

Original Datasets: We verify our proposed KD^3 on the test set of MNIST [26], CIFAR10 [24], CINIC [11], CIFAR100 [24], and TinyImageNet [25].

Webly Collected Datasets: When using MNIST as the original data, we adopt the training images from both MNIST-M [18] and SVHN [31] datasets as the webly collected data, and we grayscale the RGB images in MNIST-M and SVHN because the images in MNIST only have one channel. When using other datasets with natural images as the original data, we employ the training images from the large-scale ImageNet [12]. We also downsample the images in ImageNet to 32×32 or 64×64 to ensure the size consistency between the original data and webly collected data. Details of the adopted datasets can be found in **supplementary material**.

Implementation Details: When training on MNIST, we use Adam with the initial learning rate of 10^{-3} as the optimizer, and all student networks are trained with 40 epochs. When training on other datasets, we utilize Stochastic Gradient Descent (SGD) with weight decay of 5×10^{-4} and momentum of 0.9 as the optimizer. By following [5], all student networks are trained with 240 epochs, and the initial learning rate is set to 0.05, which is divided by ten at 150, 180, and 210 epochs. Besides, the temperature parameter τ in Eq. (15) and Eq. (16), threshold parameter V_{th} in Eq. (8), and trade-off parameter α in Eq. (18) are 0.30, 0.95, and 0.01, respectively. The parametric sensitivity will be investigated in Section 4.3.

4.1. Experiments on Image Classification Datasets

In this section, we conduct intensive experiments on five image classification tasks mentioned above to demonstrate the effectiveness of our proposed KD^3. We select four teacher-student pairs for experiments, including LeNet5-LeNet5_half [26], ResNet34-ResNet18 [22], VGGNet16-VGGNet13 [38], which are widely used in data-free methods [3, 7]. The experimental results are reported in Table 1.

Firstly, the performance of student networks trained on synthetic data is suboptimal in general, particularly when evaluating on the complex TinyImageNet, because the generated data is usually flawed or distorted. Secondly, we can observe that DFND using the instances on the Internet is still unable to produce a student network competitive to that trained on the original data, which is due to the ignorance of the distribution shift between the webly collected data and original data. In contrast, our KD^3 can successfully acquire the student networks which achieve significantly better performance than those trained on the original data in most cases. The experimental results demonstrate that our KD^3 can effectively resolve the distribution shift between the webly collected data and original data, thus training a superior student network without any original data.

Operation	Type	CIFAR10	CIFAR100
No classifier sharing	One-hot	93.42 (−1.79)	74.54 (−3.90)
	Soft	93.98 (−1.23)	76.92 (−1.52)
Instance selection	Random	90.22 (−4.99)	73.60 (−4.84)
	Only $\mathcal{N}_S$	91.99 (−3.22)	75.30 (−3.14)
	Only $\mathcal{N}_T$	94.01 (−1.20)	76.76 (−1.68)
MDCL	No $\mathcal{L}_{\text{mdcl}}$	94.48 (−0.73)	77.35 (−1.09)
	No MD	94.61 (−0.60)	77.43 (−1.01)
KD^3	$\mathcal{L}_{\text{objective}}$	**95.21**	**78.44**

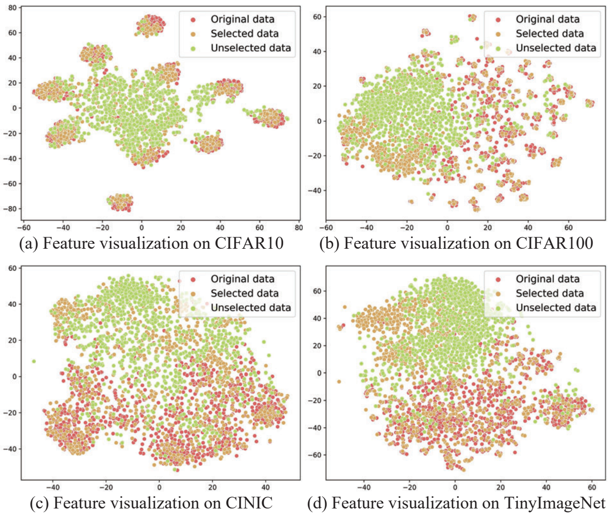
Table 2. Classification accuracy (in %) of ablation experiments. Brackets with red font denote the accuracy drop of the corresponding item compared with the complete KD^3.

4.2. Ablation Studies & Feature Visualization

Ablation Studies. We select the teacher-student pair ResNet34-ResNet18 to evaluate the three key operations in KD^3, and the results are shown in Table 2. The contribu-

Teacher	Student	#params		FLOPs		CIFAR10		CIFAR100		CINIC		TinyImageNet	
		Teacher	Student	Teacher	Student	ACC^S	KD^3	ACC^S	KD^3	ACC^S	KD^3	ACC^S	KD^3
ResNet32×4	ResNet8×4	7.41M	1.21M	1.09G	0.18G	92.09	93.05	73.09	73.17	81.74	81.71	55.40	55.13
ResNet32×4	MobileNetV2	7.41M	0.81M	1.09G	7.37M	92.38	92.16	69.06	69.40	77.61	77.95	57.15	60.60
ResNet32×4	ShuffleV1	7.41M	0.86M	1.09G	42.11M	92.92	93.24	66.43	72.15	80.13	80.93	57.94	60.01
ResNet32×4	ShuffleV2	7.41M	1.26M	1.09G	46.66M	93.23	93.53	72.60	73.14	80.64	80.74	60.93	61.41
ResNet110×2	ResNet110	6.89M	1.73M	1.02G	0.26G	93.37	94.59	74.31	73.59	84.29	84.75	59.80	60.21
ResNet110×2	ResNet116	6.89M	1.83M	1.02G	0.27G	93.21	94.57	74.46	73.75	84.45	84.68	59.85	59.52
ResNet110×2	ShuffleV1	6.89M	0.86M	1.02G	42.11M	92.92	93.24	66.43	72.15	80.13	81.25	57.94	58.54
ResNet110×2	ShuffleV2	6.89M	1.26M	1.02G	46.66M	93.23	93.46	72.60	72.93	80.64	81.54	60.93	60.80

Table 3. Classification accuracy (in %) of various network backbones. The columns "ACC^S" report the accuracies yielded by the student networks using the original data. Here, the FLOPs are calculated by feeding a 32×32 sized RGB image into the corresponding DNN.



(a) Feature visualization on CIFAR10 (b) Feature visualization on CIFAR100

(c) Feature visualization on CINIC (d) Feature visualization on TinyImageNet

Figure 3. Visualization of ResNet34-produced features by t-SNE [41]. The original images are from (a) CIFAR10, (b) CIFAR100, (c) CINIC, and (d) TinyImageNet, while the selected and unselected images are from ImageNet, each part contains 1,000 images. The data points selected by our KD^3 (*i.e.*, orange dots) show very similar distribution with the original data (*i.e.*, red dots).

tions of these key operations are analyzed as follows:

1) Classifier Sharing in Section 3.3: To estimate the effectiveness of sharing the classifier of teacher network with student network, we train a student with an initialized classifier. Moreover, to train the initialized classifier, we utilize a cross-entropy or Kullback-Leibler (KL) [23] divergence to enforce student network to mimic the one-hot predictions or soft labels of teacher network (shown in "One-hot" and "Soft"). It can be found that the performance of student network with the initialized classifier obviously degrades, which indicates that classifier sharing is vital to enhancing student's performance. It means that our method can effectively transfer the teacher-learned information of original data to student network.

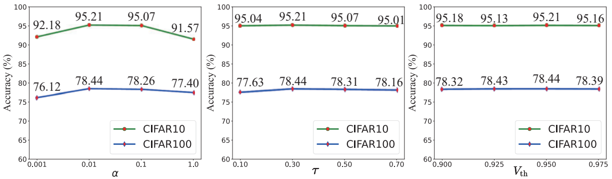
2) Instance Selection in Section 3.3: The student network obtains poor performance when the instances are randomly selected (shown in "Random") and only selected by student network (shown in "Only $\mathcal{N}_S$"). Furthermore, the student network that trained on the instances chosen by the powerful teacher network achieves relatively good performance (shown in "Only $\mathcal{N}_T$"). In particular, the student network achieves the best accuracy when utilizing the data selected by our proposed data selection method, demonstrating that our proposed data selection method can sample proper instances for student network training.

3) MixDistribution Contrastive Learning in Section 3.4: We directly remove $\mathcal{L}_{\text{mdcl}}(\mathcal{N}_S, \mathcal{N}_T)$ (shown in "No $\mathcal{L}_{\text{mdcl}}$") or replace MixDistribution by data augmentations as in [46] (shown in "No MD") to train student network. The accuracy of student network has reduced significantly when evaluated on test data of which the distribution is different from the webly collected data. The results demonstrate that MixDistribution contrastive learning is critical to solving the distribution shift problem.

Visualization of Features. To further understand the effectiveness of our proposed data selection method, we visualize the ResNet34-provided features of images from original data, selected data, and unselected data. The original training images are provided by CIFAR10, CIFAR100, CINIC, and TinyImageNet, and webly collected images are from ImageNet. The t-SNE [41] visualization results are shown in Fig. 3, from which we can observe that the distributions of selected images are close to the original images in feature space. The visualization results demonstrate that our data selection method can effectively select the webly collected instances with the similar distribution to original data. More visualization results are shown in **supplementary material**.

4.3. Parametric Sensitivity

The tuning parameters in our KD^3 include the trade-off parameter α in Eq. (18), temperature parameter τ in Eq. (15) and Eq. (16), and threshold parameter V_{th} in Eq. (8). This section analyzes the sensitivity of our KD^3 to these parameters on the CIFAR dataset. The ResNet34 and ResNet18 are selected as teacher and student, respec-



(a) Analysis of α (b) Analysis of τ (c) Analysis of V_{th}

Figure 4. Parametric sensitivity of (a) α in Eq. (18), (b) τ in Eq. (15) and Eq. (16), and (c) V_{th} in Eq. (6).

Teacher	Student	Dataset	5%	25%	50%	75%	100%
ResNet34	ResNet18	CIFAR10	84.35	92.83	94.11	94.97	95.21
		CIFAR100	61.17	75.57	77.02	77.85	78.44
		CINIC	75.01	82.19	84.49	85.88	86.55
		TinyImageNet	52.28	63.87	65.86	66.15	66.24
ResNet34	VGGNet13	CIFAR10	82.90	92.66	93.76	94.18	94.28
		CIFAR100	53.41	73.99	76.06	76.18	77.92
		CINIC	71.06	82.00	83.39	83.76	84.07
		TinyImageNet	50.71	60.01	62.99	63.54	63.83

Table 4. Classification accuracies (in %) of student networks under different webly collected data ratios.

tively. We examine the resulting accuracy during training by changing one parameter while holding the others.

Fig. 4 depicts the curves of test accuracy for student network when the parameters vary. The parameters α, τ, and V_{th} vary within $\{0.001, 0.01, 0.1, 1\}$, $\{0.1, 0.3, 0.5, 0.7\}$, and $\{0.900, 0.925, 0.950, 0.975\}$, respectively. Even though these parameters vary over a wide range, we can obverse that the curves of accuracy are generally smooth and relatively stable, which indicates that the performance of student network is robust to the variations of parameters. Therefore, the parameters in our KD^3 are easy to tune.

4.4. Extented Experiments

In this section, we conduct intensive extented to further verify the performance of KD^3.

Experiments on More Network Backbones. We conduct experiments on four benchmark datasets to further verify the performance of KD^3 equipped with various widely-used teacher-student pairs [22, 29, 37, 47]. The results are reported in Table 3. It can be found that the student networks trained by our KD^3 consistently achieve competitive performance to those trained on the original data, even though some student networks are with different styles of the teacher network. The experimental results demonstrate that our data-free method KD^3 can be flexibly employed to teacher-student pairs with various structures to train reliable student networks.

Experiments on Part Webly Collected Data. We further conduct intensive experiments to evaluate the performance when the ratio of available webly collected data ranges from 5%∼100%. The experimental results are shown in Table 4, we can observe that our method still obtains satisfactory performance by using a small part of web data. Meanwhile, we find that the performance of student network can be consistently improved with the increase of the available webly collected data.

5. Conclusion

This paper proposed a new data-free approach termed KD^3 to train student networks using the webly collected data. To our best knowledge, we are the first to address the commonly overlooked yet important distribution shift issue between the webly collected data and original data in knowledge distillation. Our proposed KD^3 adopts three main techniques to tackle such distribution shift, namely: 1) selection of webly collected instances with the similar distribution to original data; 2) alignment of feature distributions between the teacher network and student network with parameter-shared classifiers; and 3) promotion of feature consistency for input instances and MixDistribution-generated instances. Intensive experiments demonstrated that our KD^3 can effectively handle the distribution shift to train reliable student networks without using the original training data.

Acknowledgment

C.G. was supported by NSF of China (No: 61973162), NSF of Jiangsu Province (No: BZ2021013), NSF for Distinguished Young Scholar of Jiangsu Province (No: BK20220080), the Fundamental Research Funds for the Central Universities (Nos: 30920032202, 30921013114), CAAI-Huawei MindSpore Open Fund, and "111" Program. M.S. was supported by the Institute for AI and Beyond, The University of Tokyo.

References

[1] Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, 2014. 2

[2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *International Conference on Learning Representations (ICLR)*. 3

[3] Kuluhan Binici, Shivam Aggarwal, Nam Trung Pham, Karianto Leman, and Tulika Mitra. Robust and resource-efficient data-free knowledge distillation by generative pseudo replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6089–6096, 2022. 7

[4] Akshay Chawla, Hongxu Yin, Pavlo Molchanov, and Jose Alvarez. Data-free knowledge distillation for object detection. In *Proceedings of the IEEE/CVF Winter Conference on*

Applications of Computer Vision (WACV), pages 3289–3298, 2021. 3

[5] Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. Knowledge distillation with the reused teacher classifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11933–11942, 2022. 2, 7

[6] Hanting Chen, Tianyu Guo, Chang Xu, Wenshuo Li, Chunjing Xu, Chao Xu, and Yunhe Wang. Learning student networks in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6428–6437, 2021. 1, 3, 7

[7] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3514–3522, 2019. 1, 3, 6, 7

[8] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5008–5017, 2021. 3

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning (ICML)*, pages 1597–1607. PMLR, 2020. 3, 6

[10] Yoojin Choi, Jihwan Choi, Mostafa El-Khamy, and Jungwon Lee. Data-free network quantization with adversarial knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, pages 710–711, 2020. 1, 3, 7

[11] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018. 7

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. 1, 7

[13] Kien Do, Hung Le, Dung Nguyen, Dang Nguyen, Haripriya Harikumar, Truyen Tran, Santu Rana, and Svetha Venkatesh. Momentum adversarial distillation: Handling large distribution shifts in data-free knowledge distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 1

[14] Yuntao Du, Haiyang Yang, Mingcai Chen, Juan Jiang, Hongtao Luo, and Chongjun Wang. Generation, augmentation, and alignment: A pseudo-source domain based method for source-free domain adaptation. *arXiv preprint arXiv:2109.04015*, 2021. 3, 5

[15] Gongfan Fang, Jie Song, Chengchao Shen, Xinchao Wang, Da Chen, and Mingli Song. Data-free adversarial distillation. *arXiv:1912.11006*, 2019. 3, 6

[16] Gongfan Fang, Jie Song, Xinchao Wang, Chengchao Shen, Xingen Wang, and Mingli Song. Contrastive model inversion for data-free knowledge distillation. *arXiv preprint arXiv:2105.08584*, 2021. 1, 3, 7

[17] Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. Rethinking importance weighting for deep learning under distribution shift. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:11996–12007, 2020. 3

[18] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research (JMLR)*, 17(1):2096–2030, 2016. 7

[19] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 1

[20] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 3

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 1

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 7, 9

[23] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015. 1, 2, 3, 8

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 7

[25] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 7

[26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 7

[27] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *International Conference on Machine Learning (ICML)*, pages 6028–6039. PMLR, 2020. 5

[28] Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2736–2744, 2017. 1

[29] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 9

[30] Paul Micaelli and Amos J Storkey. Zero-shot knowledge transfer via adversarial belief matching. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 3, 7

[31] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 7

[32] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 22(10):1345–1359, 2010. 3

[33] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5007–5016, 2019. 2, 6

[34] Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008. 3

[35] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, pages 525–542. Springer, 2016. 1

[36] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv:1412.6550*, 2014. 2, 3

[37] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018. 9

[38] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2014. 7

[39] Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Müller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research (JMLR)*, 8(5), 2007. 3

[40] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1365–1374, 2019. 2, 6

[41] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research (JMLR)*, 9(11), 2008. 8

[42] Qin Wang, Wen Li, and Luc Van Gool. Semi-supervised learning by augmented distribution alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1466–1475, 2019. 3

[43] Nicolai Wojke and Alex Bewley. Deep cosine metric learning for person re-identification. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 748–756. IEEE, 2018. 6

[44] Yixing Xu, Yunhe Wang, Hanting Chen, Kai Han, Chunjing Xu, Dacheng Tao, and Chang Xu. Positive-unlabeled compression on the cloud. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2565–2574, 2019. 3

[45] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8715–8724, 2020. 1, 3, 6

[46] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv:1710.09412*, 2017. 8

[47] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6848–6856, 2018. 9

[48] Yiman Zhang, Hanting Chen, Xinghao Chen, Yiping Deng, Chunjing Xu, and Yunhe Wang. Data-free knowledge distillation for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7852–7861, 2021. 3

[49] Ziyi Zhang, Weikai Chen, Hui Cheng, Zhen Li, Siyuan Li, Liang Lin, and Guanbin Li. Divide and contrast: Source-free domain adaptation via adaptive contrastive learning. *arXiv preprint arXiv:2211.06612*, 2022. 3

[50] Haoran Zhao, Xin Sun, Junyu Dong, Milos Manic, Huiyu Zhou, and Hui Yu. Dual discriminator adversarial distillation for data-free model compression. *International Journal of Machine Learning and Cybernetics (IJMLC)*, 13(5):1213–1230, 2022. 1, 3, 6

[51] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *International Conference on Learning Representations (ICLR)*, 2021. 6

[52] Jinguo Zhu, Shixiang Tang, Dapeng Chen, Shijie Yu, Yakun Liu, Mingzhe Rong, Aijun Yang, and Xiaohua Wang. Complementary relation contrastive distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269, 2021. 6