Learning Student Network Under Universal Label Noise

Jialiang Tang, *Student Member, IEEE*, Ning Jiang, *Member, IEEE*, Hongyuan Zhu, *Member, IEEE*, Joey Tianyi Zhou, *Senior Member, IEEE*, and Chen Gong, *Senior Member, IEEE*

***Abstract*— Data-free knowledge distillation aims to learn a small student network from a large pre-trained teacher network without the aid of original training data. Recent works propose to gather alternative data from the Internet for training student network. In a more realistic scenario, the data on the Internet contains two types of label noise, namely: 1) *closed-set* label noise, where some examples belong to the known categories but are mislabeled; and 2) *open-set* label noise, where the true labels of some mislabeled examples are outside the known categories. However, the latter is largely ignored by existing works, leading to limited student network performance. Therefore, this paper proposes a novel data-free knowledge distillation paradigm by utilizing a webly-collected dataset under *universal* label noise, which means both closed-set and open-set label noise should be tackled. Specifically, we first split the collected noisy dataset into clean set, closed noisy set, and open noisy set based on the prediction uncertainty of various data types. For the closed-set noisy examples, their labels are refined by teacher network. Meanwhile, a noise-robust hybrid contrastive learning is performed on the clean set and refined closed noisy set to encourage student network to learn the categorical and instance knowledge inherited by teacher network. For the open-set noisy examples unexplored by previous work, we regard them as unlabeled and conduct self-supervised learning on them to enrich the supervision signal for student network. Intensive experimental results on image classification tasks demonstrate that our approach can achieve superior performance to state-of-the-art data-free knowledge distillation methods.**

***Index Terms*— Data-free knowledge distillation, universal label noise, self-supervised learning, model compression.**

Manuscript received 23 October 2022; revised 20 November 2023 and 1 May 2024; accepted 3 July 2024. Date of publication 29 July 2024; date of current version 1 August 2024. The work of Ning Jiang was supported in part by Sichuan Science and Technology Program under Grant 2022YFG0324 and in part by the Nuclear Medicine Artificial Intelligence Research Center of Fujiang Laboratory. The work of Hongyuan Zhu was supported in part by the Agency for Science, Technology and Research (A*STAR) AME Programmatic Funding under Grant A18A2b0046, in part by the RobotHTPO Seed Fund under Project C211518008, and in part by the EDB Space Technology Development Grant under Project S22-19016-STDP. The work of Joey Tianyi Zhou was supported in part by the National Research Foundation, Prime Minister's Office, Singapore; in part by the Ministry of Communications and Information, under its Online Trust and Safety (OTS) Research Program under Grant MCI-OTS-001; and in part by the SERC Central Research Fund (Use-Inspired Basic Research). The work of Chen Gong was supported in part by the NSF of China under Grant 62336003 and Grant 12371510, in part by the NSF of Jiangsu Province under Grant BZ2021013, in part by the NSF for Distinguished Young Scholar of Jiangsu Province under Grant BK20220080, and in part by the Fundamental Research Funds for the Central Universities under Grant 30920032202 and Grant 30921013114. The associate editor coordinating the review of this article and approving it for publication was Dr. Marie Chabert. *(Corresponding author: Chen Gong.)*

Jialiang Tang and Chen Gong are with the Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education and Jiangsu Key Laboratory of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: tangjialiang@njust.edu.cn; chen.gong@njust.edu.cn).

Ning Jiang is with the School of Computer Science and Technology, Southwest University of Science and Technology, Mianyang 621010, China (e-mail: jiangning@swust.edu.cn).

Hongyuan Zhu is with the Institute for Infocomm Research (I^2R) and the Centre for Frontier AI Research (CFAR), Agency for Science, Technology and Research (A*STAR), Singapore 138632 (e-mail: zhuh@i2r.a-star.edu.sg).

Joey Tianyi Zhou is with the Centre for Frontier A Research (CFAR), Institute of High Performance Computing (IHPC), Agency for Science Technology and Research (A*STAR), Singapore 138632, and also with the Centre for Advanced Technologies in Online Safety (CATOS), Singapore 138632 (e-mail: zhouty@cfar.a-star.edu.sg).

Digital Object Identifier 10.1109/TIP.2024.3430539

I. Introduction

DEEP Neural Networks (DNNs) have shown very impressive performance in various computer vision tasks [18], [48]. However, the capacity of DNNs is usually quite large, which seriously hinders their deployment on some embedded devices such as smartphones and security cameras. To enable the application of DNNs on these practical resource-limited devices, numerous works [11], [19], [33] have been done to compress pre-trained large DNNs to small ones. Among these methods, knowledge distillation [19] has shown very encouraging results, which transfers information from the original large network (*a.k.a.* teacher network) to a small portable network (*a.k.a.* student network) to achieve model compression.

Existing model compression algorithms based on knowledge distillation can usually achieve a large compression ratio without dramatic performance loss when the original training data for teacher network is available. However, the original data might usually be untouchable due to various practical limitations, such as data management considerations and privacy issues. For instance, the large-scale image classification dataset ImageNet [10] contains 14,197,122 images and requires about 155*GB* of memory space to store, which is too "heavy" to share among different machines. Besides, some types of data, like people's daily photos and clinical records, are private, and distributing these data may incur legal problems. Therefore, some recent distillation works focus on data-free methods, which aim to reconstruct images to compress large pre-trained networks without the aid of original training data. For example, Lopes et al. [34] leverage the "meta-data" provided by a pre-trained model to approximate the original data. Data-Free Learning (DFL) [6] introduces a generator to compose images under the supervision of teacher network. DeepInversion [56] utilizes the statistics stored in the middle layers of the teacher

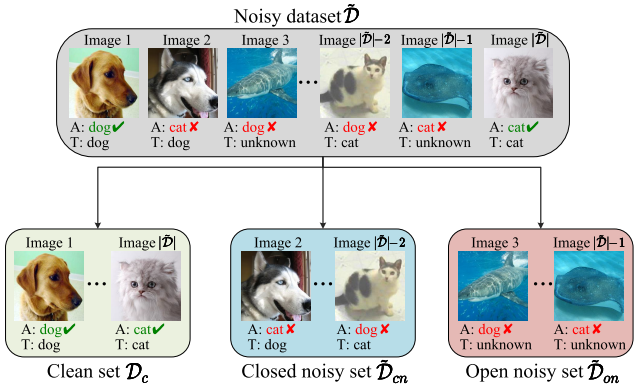


Fig. 1. The illustration of collected noisy dataset $\tilde{\mathcal{D}}$ with $|\tilde{\mathcal{D}}|$ image examples, where we assume that the original dataset contains two categories including "cat" and "dog". "A" and "T" represent the annotated label in the dataset and true label of the corresponding image, respectively. The noisy dataset can be divided into clean set $\mathcal{D}_c$, closed noisy set $\tilde{\mathcal{D}}_{cn}$, and open noisy set $\tilde{\mathcal{D}}_{on}$. The clean examples in $\mathcal{D}_c$ are all annotated correctly. For the closed-set noisy examples in $\tilde{\mathcal{D}}_{cn}$, the images of "dog" ("cat") are mislabeled as "cat" ("dog"). For the open-set noisy examples in $\tilde{\mathcal{D}}_{on}$, these images are incorrectly annotated as "cat" or "dog" but do not belong to any of the two categories.

to synthesize data. Even though these approaches can produce fake data and perform model compression, due to the intrinsic discrepancy between the generated image and real image, the capability of compressed models is still limited to some extent.

Therefore, instead of generating fake data, there are also some methods targeting to leverage massive yet noisy data on the Internet. For example, Chen et al. [5] propose Data-Free Noisy Distillation (DFND) to utilize data in the wild to compress a student network from the pre-trained teacher network. However, they only consider the *closed-set* label noise, which means that the ground-truth label of noisy image examples still falls into the known classes-of-interests. Unfortunately, in a more realistic scenario, since we do not know the real labels of collected images, it is quite possible that some of the images possess true class labels that are not present in the original training data (see Fig. 1). Therefore, such *open-set* label noise is also ubiquitous in the wild and should be taken into consideration during the distillation process. For example, when we are interested in classifying different fruits and type "apple" into an image search engine, it is possible to get the images of fruit as well as the iPhone with "apple" brand. Apparently, the obtained iPhone images are unexpected out-of-distribution examples, which constitute open-set noisy data. In this sense, we claim that the realistic situations usually contain *universal* label noise, which includes both traditional closed-set label noise and unexplored open-set label noise, while the latter is largely ignored by the current distillation approaches.

Based on the above consideration, in this paper, we propose a new method named "**MO**del **D**istillation with **U**niversal **L**abel noise" (MODUL) to distill a teacher network to a student network on the data with universal label noise in the wild to resolve the data-free knowledge distillation problem. More specifically, we first divide the collected image set into clean set (*i.e.*, $\mathcal{D}_c$), closed noisy set (*i.e.*, $\tilde{\mathcal{D}}_{cn}$), and open noisy set (*i.e.*, $\tilde{\mathcal{D}}_{on}$) based on the output loss values incurred by the contained image examples. Then, for the images within clean set $\mathcal{D}_c$, we directly retain their original annotated labels for training. For the images within closed noisy set $\tilde{\mathcal{D}}_{cn}$, they are re-annotated by the pre-trained teacher network. For the images within open noisy set $\tilde{\mathcal{D}}_{on}$, one simple way is to directly discard them as they seem to be irrelevant to the core task at first glance. However, we argue that they also contain meaningful information and are still helpful if they are appropriately utilized. Since their labels are noisy, we may treat them as unlabeled and utilize self-supervised learning to explore their precious feature information. Specifically, we may introduce rotation recognition task, which is a popular self-supervised learning strategy, to promote student network to learn rotation-invariant representation for each image, thereby enhancing the representation ability and learning performance of student network. Similarly, we also conduct the hybrid contrastive learning on the images in clean set and refined closed noisy set to transfer categorical and instance knowledge from teacher network to student network to further boost the representation ability of student network. Thanks to the proper usage of the above three types of image sets, an effective and portable student model can be learned, as shown in Fig. 2. Intensive experimental results show that our MODUL can train a superior student network when compared with state-of-the-art methods, including [5], [6], [12], [37], and [56]. The accuracy of the small student network is also comparable to that of the large teacher network.

The contributions of our MODUL are summarized as follows:

1) We propose a new data-free knowledge distillation method termed MODUL to compress pre-trained teacher networks through the noisy data with universal label noise collected from the Internet. Our MODUL can effectively process closed-set and open-set label noise, leading to high-performance compact student networks.
2) We develop a hybrid contrastive learning loss to transfer category-level and instance-level knowledge from teacher network to student network, which significantly improves the performance of student network.
3) We properly tackle the universal label noise and obtain a noise-robust student network. In particular, we innovatively regard the open-set noisy examples as unlabeled and explore them by self-supervised learning.

The overall structure of this paper is presented below. Section II reviews the related works. Section III describes the traditional knowledge distillation. Furthermore, Section IV details the implementation of our designed method, and Section V shows the experimental results. Finally, Section VI concludes the entire paper.

II. Related Works

In this section, we review the related works of this paper, including model compression, learning with noisy labels, and contrastive learning.

A. Model Compression

Network compression aims to convert a large-size deep neural network to a small one. So far, various strategies [11],

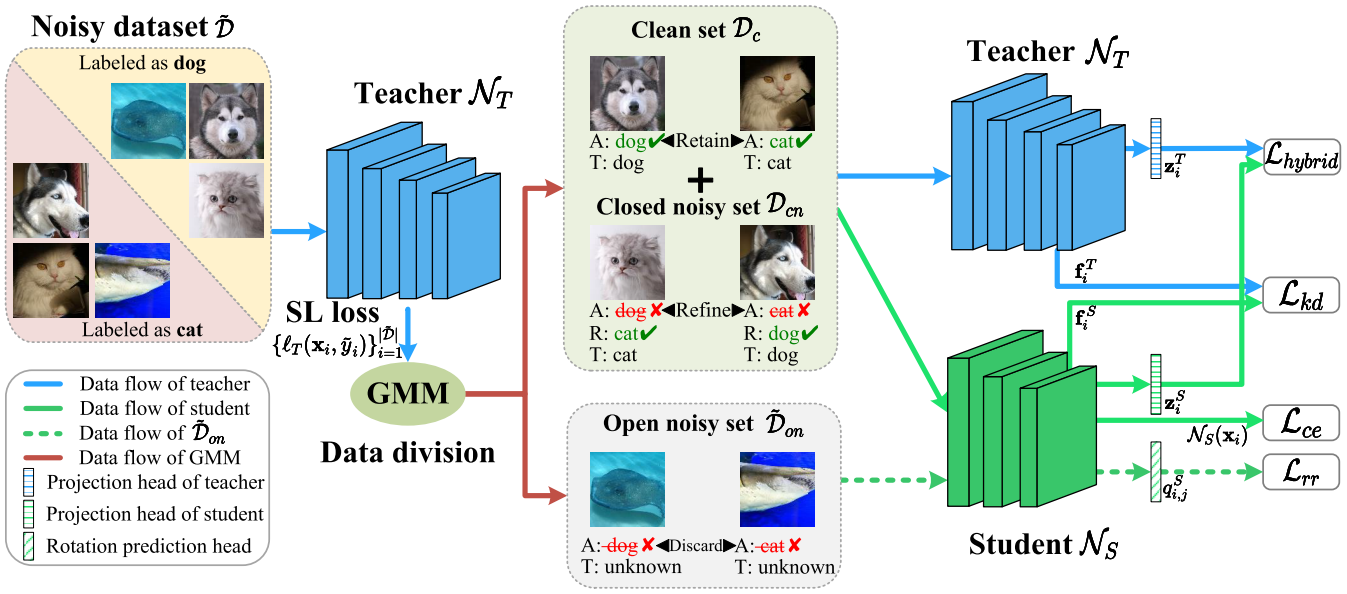


Fig. 2. The diagram of our MODUL method. The collected noisy dataset $\tilde{\mathcal{D}}$ is composed of two categories of "dog" and "cat", and we define the annotated label, true label, and refined label of the corresponding image as "A", "T", and "R", respectively. The $\tilde{\mathcal{D}}$ is first divided into clean set $\mathcal{D}_c$, closed noisy set $\mathcal{D}_{cn}$, and open noisy set $\tilde{\mathcal{D}}_{on}$ by teacher network $\mathcal{N}_T$ and Gaussian Mixture Model (GMM). Then, a hybrid contrastive learning loss (*i.e.*, $\mathcal{L}_{hybrid}$) is conducted on $\mathcal{D}_c$ and $\mathcal{D}_{cn}$ to transfer categorical and instance knowledge from teacher network to student network $\mathcal{N}_S$. Moreover, rotation recognition (*i.e.*, $\mathcal{L}_{rr}$) is implemented on $\tilde{\mathcal{D}}_{on}$ to further enhance student's representation ability. Besides, $\mathcal{D}_c$ and $\mathcal{D}_{cn}$ are jointly used to train student network through standard back-propagation (*i.e.*, $\mathcal{L}_{ce}$) and conduct knowledge distillation (*i.e.*, $\mathcal{L}_{kd}$).

[33], [41], [45] have been developed for achieving network compression, such as network pruning, network quantization, and knowledge distillation. Since this paper cares about knowledge distillation, we will review the representative works in this area in the following.

In knowledge distillation, a shallow and narrow student network is usually learned by receiving the knowledge of a deep and wide pre-trained teacher network. Based on the types of transferred knowledge, existing methods can generally be classified into three categories, namely response-based methods [2], [19], feature-based methods [41], [58], and relation-based methods [39], [55]. Response-based distillation approaches directly utilize the outputs of the last classification layer of teacher network for teaching the student network and encourage the student to simulate the predictions of the teacher. For example, Ba et al. [2] require a shallow student network to mimic the outputs before the softmax of a deep teacher network by minimizing a ℓ_2 loss. In contrast, Hinton et al. [19] propose to input the predictions of teacher network into the softmax with a temperature parameter to calculate the soft labels, which helps to reserve the category correlations computed by the teacher model. Recently, Zhao et al. [61] decouple the soft label as target knowledge (*i.e.*, the prediction for target class) and non-target knowledge (*i.e.*, relationship between other categories), which effectively improves the performance of response-based distillation.

Feature-based methods extract the output features from the bottom or middle layers of teacher network for teaching the student network, with a consideration that the feature maps of DNNs are informative and beneficial for knowledge transfer. Therefore, Romero et al. [41] introduce the intermediate features of teacher network as hints, and student network simply mimics the hints by minimizing the mean-square error loss. Subsequently, various methods [4], [58] are developed to explore how to extract and transfer features efficiently. For instance, Zagoruyko and Komodakis [58] find that directly using the feature maps of teacher is inefficient and refine the features from the middle layers of teacher network by attention mechanism. Recently, Chen et al. [4] propose to collect features from multiple layers of teacher network to learn each layer of student network.

Relation-based approaches incline to exploit the teacher-student relationships between different layers and examples during teaching, which deviate from the previous works that enforce the student network to learn the outputs (*e.g.*, soft labels or important features) of teacher network. To explore the relationship between layers, Yim et al. [55] calculate the inner product of the features from two layers of the teacher network. To deploy the connections between examples, relational knowledge distillation [39] calculates the Euclidean distance between every two images and the angle of every three images for distillation. Besides, [54] explores pixel-to-pixel and pixel-to-region relations over images to help learn a student network for semantic segmentation tasks.

The above methods can compress various DNNs on different datasets. However, these methods cannot deal with the data-free knowledge distillation as mentioned in the introduction. Therefore, some researchers attempt to reconstruct data from the pre-trained models to compress DNNs for achieving data-free knowledge distillation. For instance, Lopes et al. [34] propose to reproduce the data from the "meta-data" in teacher network to train student network. Data-Free Learning (DFL) [6] introduces a generator and utilizes teacher network to supervise it to build the fake data that conform to the distribution of the teacher's original training data. DeepInversion [56] inverts the means and variances of the

features of teacher network to synthesize images from random noise. Although these methods can successfully imitate data and compress deep models, the performance of the compressed models is still suboptimal due to the essential gap between the generated data and actual data.

Therefore, instead of approximating the original data, Chen et al. [5] propose to use the plentiful but noisy data from the wild to train the student network and effectively improve the model performance when lacking the original data. Unfortunately, they only consider the closed-set noisy examples and neglect more common open-set noisy examples contained by the collected data, which will deteriorate the performance of student network. In contrast, in this work, we will consider different types of label noise in the wild to train an improved student network when the original training data is not available.

B. Learning With Noisy Labels

Recently, there has been increasing attention on designing robust classifiers for dealing with noisy labels, which are mainly based on the following four categories, including loss correction [40], [46], [53], robust loss design [14], [15], sample selection [16], [22], [57], and optimal transport [9], [13].

Loss correction methods eliminate the adverse influence of noisy labels by deploying a noise transition matrix. For example, Sukhbaatar et al. [46] learn a noise transition matrix by inserting a learnable noise layer at the top of the network to match the noisy label distribution. Patrini et al. [40] employ a small set of clean data to estimate the noise transition matrix for training a robust model. Moreover, Xia et al. [53] develop a way for estimating the noise transition matrix without the aid of clean data.

Some approaches aim to design an inherently robust loss function to combat noisy labels. For example, Ghosh et al. [15] propose to learn a model by Mean-Absolute Error (MAE) loss on noisy data and prove that MAE is more robust to noisy labels than Cross Entropy (CE) loss. Feng et al. [14] reveal the correlations between MAE and CE and propose the Taylor CE loss, which can adjust the fitting degree for the training labels based on the order of Taylor series.

Sample selection methods devote to picking up possibly clean data for training in each iteration. Jiang et al. [22] develop a dynamic curriculum for the neural network, where "simple" examples are firstly chosen for network training. Inspired by [22], Co-teaching [16] and Co-teaching+ [57] train two deep networks alternately to avoid error accumulation during the process of clean data selection. Similarly, Wei et al. [51] solve the problem by training two different networks to compute the joint loss to select clean examples with small loss values.

Besides, optimal transport methods are also powerful in addressing label noise. Among them, Classification Loss with Entropic Optimal Transport (CLEOT) [9] regularizes the classification model by entropic regularization since it successfully trains a reliable model on noisy datasets. Wasserstein Adversarial Regularization (WAR) [13] utilizes the similarity between classes to regularize the model predictions, which are evaluated by the Wasserstein distance.

Above-mentioned works mainly focus on processing the closed-set noisy labels, which assume that the true label of an image is among the known classes of the training data. To deal with open-set label noise, Wang et al. [49] apply a Siamese network to iteratively identify the open-set noisy labels and explore the deep discriminative features to impel noisy data away from clean data. Sachdeva et al. [42] propose a general framework to learn with combined closed-set and open-set noisy labels, which first trains a network NetS to divide the noisy data and then another network NetD for learning on the separated data. To the best of our knowledge, we are the first to implement data-free knowledge distillation under the universal label noise with both closed-set and open-set noisy labels.

C. Contrastive Learning

Contrastive Learning (CL) [8], [29], [36] is a mainstream approach in self-supervised learning [1] to learn robust representations for practical downstream tasks. In general, CL can be unsupervised or supervised. Unsupervised CL algorithms [17], [52] divide data as positive and negative pairs and maximize (minimize) similarities between positive (negative) pairs in the representation space. In unsupervised CL, there is usually only one positive pair composed of two transformations of the same instance, and different instances are used to construct a set of negative pairs. Here, the number of negative examples usually surpasses that of positive examples, which has been demonstrated to benefit DNNs in learning strong representations [8], [17]. Recent works [24], [28] have extended CL to the fully-supervised setting to leverage the label information of the dataset. Unlike unsupervised CL [17] which uses one positive pair and many negative pairs among data points, supervised CL [24], [28] considers a series of positive and negative pairs, among which positive pairs are formed by the images within the same class and negative pairs are selected from the images of different classes. In this study, to improve the representation ability of student network, we train student network on collected noisy data via the well-designed contrastive learning algorithms.

III. Conventional Knowledge Distillation

Conventional knowledge distillation methods [19], [41] learn a compact student network $\mathcal{N}_S$ from a large pre-trained teacher network $\mathcal{N}_T$. Given $\mathcal{X} \in \mathbb{R}^d$ (d means the dimensionality) as the input feature space, and $\mathcal{Y} \in \{1, \ldots, K\}$ (K represents the total number of known classes) as the label space of classes-of-interests, the student network is trained on the original training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|\mathcal{D}|} \in \mathcal{X} \times \mathcal{Y}$ with size $|\mathcal{D}|$ ("$|\cdot|$" represents the cardinality of corresponding set throughout this paper), where all labels $\{y_i\}_{i=1}^{|\mathcal{D}|}$ of images $\{\mathbf{x}_i\}_{i=1}^{|\mathcal{D}|}$ are correctly provided. For a batch of data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{b} \in \mathcal{D}$ (b denotes the batchsize of corresponding data throughout this paper), the loss function for knowledge distillation is formulated as:

$$\mathcal{L}_{kd} = \frac{1}{b}\sum_{i=1}^{b} \lambda \mathcal{H}_{ce}\left(\mathcal{N}_S(\mathbf{x}_i), y_i\right) + (1-\lambda)\mathcal{H}_{kt}\left(\mathbf{f}_i^S, \mathbf{f}_i^T\right), \quad (1)$$

where the variables with superscripts "T" and "S" denote that they are output by teacher network and student network, respectively; $\mathcal{H}_{ce}$ indicates the cross entropy loss that encourages the student network to learn from the data; $\mathcal{H}_{kt}$ is the knowledge transfer function (*e.g.*, Kullback-Leibler divergence or Euclidean distance) that promotes the student to mimic the teacher's output $\mathbf{f}_i^T$ (*e.g.*, soft label or feature map); and $\lambda > 0$ is the trade-off parameter to balance the two terms.

However, as stated in Section I, traditional knowledge distillation cannot work well if the original training set $\mathcal{D}$ is absent. Therefore, a series of works have been done to study data-free knowledge distillation [5], [6], [34], which is also the target of this paper.

IV. Our Approach

In our problem, there is only a pre-trained teacher network $\mathcal{N}_T$ and its original training data $\mathcal{D}$ is inaccessible. Therefore, by following [5], we may collect massive noisy data $\tilde{\mathcal{D}} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^{|\tilde{\mathcal{D}}|} \in \mathcal{X} \times \mathcal{Y}$ from the Internet to help train the small student network $\mathcal{N}_S$, where the notations with superscript "$\sim$" indicates that they are noisy. Since $\tilde{\mathcal{D}}$ is directly acquired from the Internet, the image $\mathbf{x}_i \in \tilde{\mathcal{D}}$ may be mislabeled to label $\tilde{y}_i \in \mathcal{Y}$ that deviates from its ground-truth label y_i. Consequently, our target is to obtain a small $\mathcal{N}_S$ that can predict label for any unseen $\mathbf{x}$ that with a true label $y \in \mathcal{Y}$ based on $\mathcal{N}_T$ and $\tilde{\mathcal{D}}$. For the label noise within $\tilde{\mathcal{D}}$, we simultaneously consider both closed-set noise and open-set noise, which we call "universal noise" in this paper and are ubiquitous in practice. For closed-set noise, it means that an image $\mathbf{x}$ has ground-truth label $y \in \mathcal{Y}$ but is mislabeled as $\tilde{y} \in \mathcal{Y}$. Such noise will introduce erroneous gradients in training the student network. For open-set noise, it means that an image $\mathbf{x}$ is incorrectly labeled as $\tilde{y} \in \mathcal{Y}$ but its actual label $y \notin \mathcal{Y}$. Such noise will bring useless information to student network during standard back-propagation and knowledge transfer.

Formally, we define two noise ratios as $\rho_1, \rho_2 \in [0, 1]$. More specifically, $\rho_1 = \frac{\#\{(\mathbf{x},\tilde{y})|\tilde{y}\neq y\}}{|\tilde{\mathcal{D}}|}$ denotes the proportion of noisily-labeled examples in the noisy dataset $\tilde{\mathcal{D}}$ and $\rho_2 = \frac{\#\{(\mathbf{x},\tilde{y})|\tilde{y}\neq y, y\notin\mathcal{Y}\}}{\#\{(\mathbf{x},\tilde{y})|\tilde{y}\neq y\}}$ represents the ratio of the examples with open-set noisy labels in all noisily-labeled examples, where # denotes the number of elements in the corresponding set that satisfy the given condition. Therefore, there are $|\mathcal{D}_c| = |\tilde{\mathcal{D}}| \times (1 - \rho_1)$ images belonging to clean set $\mathcal{D}_c$ with correct label $\tilde{y} = y$. For the remaining $|\tilde{\mathcal{D}}| \times \rho_1$ images with noisy labels, there are $|\tilde{\mathcal{D}}_{on}| = |\tilde{\mathcal{D}}| \times \rho_1 \times \rho_2$ images in open noisy set $\tilde{\mathcal{D}}_{on}$, and $|\tilde{\mathcal{D}}_{cn}| = |\tilde{\mathcal{D}}| \times \rho_1 \times (1 - \rho_2)$ images are in closed noisy set $\tilde{\mathcal{D}}_{cn}$. That is to say, $\tilde{\mathcal{D}} = \mathcal{D}_c \cup \tilde{\mathcal{D}}_{cn} \cup \tilde{\mathcal{D}}_{on}$ and $|\tilde{\mathcal{D}}| = |\mathcal{D}_c| + |\tilde{\mathcal{D}}_{cn}| + |\tilde{\mathcal{D}}_{on}|$. Note that $\mathcal{D}_c$, $\tilde{\mathcal{D}}_{cn}$, and $\tilde{\mathcal{D}}_{on}$ are unknown before distillation and they should be identified by our designed algorithm.

To solve the above-mentioned problem and train a compact student on the noisy dataset $\tilde{\mathcal{D}}$, here we propose an effective data-free knowledge distillation framework termed MODUL. As shown in Fig. 2, our MODUL contains three key steps, namely: 1) noisy data division, which explores the loss characteristics for different types of data to divide the noisy dataset $\tilde{\mathcal{D}}$ to $\mathcal{D}_c$, $\tilde{\mathcal{D}}_{cn}$, and $\tilde{\mathcal{D}}_{on}$; 2) knowledge distillation, which utilizes the in-distribution images in $\mathcal{D}_c$ and $\tilde{\mathcal{D}}_{cn}$ to transfer categorical and instance knowledge from teacher to student; 3) noisy data handling, which effectively handles the noisy examples in $\mathcal{D}_c$, $\tilde{\mathcal{D}}_{cn}$, and $\tilde{\mathcal{D}}_{on}$, leading to a robust and powerful student network. Next, we detail these key steps in Sections IV-A, IV-B, and IV-C, respectively. The designed loss function and algorithm implementation will be introduced in Section IV-D.

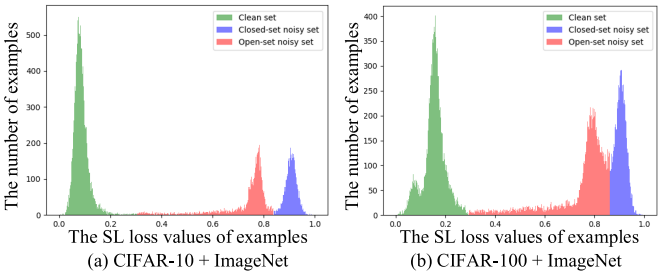


Fig. 3. Distribution of SL loss values of examples in the noisy dataset $\tilde{\mathcal{D}}$ with noise ratios $\rho_1 = 0.50$ and $\rho_2 = 0.50$. The clean examples and closed-set noisy examples are extracted from CIFAR-10 or CIFAR-100 [25] dataset, and the open-set noisy examples are sampled from the ImageNet [10] dataset.

A. Noisy Data Division

The available noisy dataset $\tilde{\mathcal{D}}$ collected from the Internet potentially contains three types of image sets, namely clean set $\mathcal{D}_c$, closed noisy set $\tilde{\mathcal{D}}_{cn}$, and open noisy set $\tilde{\mathcal{D}}_{on}$. Therefore, it is crucial to identify images in $\tilde{\mathcal{D}}$ into corresponding sets to prevent them from hurting the student network training. To solve these problems, we explore the loss value distribution of different data types. Fig. 3 shows the distribution of per-example subjective logic (SL) loss [43] (will be detailed later) on CIFAR-10 (CIFAR-100) [25] and ImageNet [10] datasets. We can see that the classifications of the teacher network on the images with clean labels in $\mathcal{D}_c$ tend to be "confident", therefore producing small loss values. In contrast, **closed-set** noisy examples can be classified into a certain category with high probability by the teacher network. Nevertheless, since they are mislabeled, they will incur large loss values. Meanwhile, **open-set** noisy examples do not belong to any of the classes-of-interests, and the teacher network usually makes "blurry" predictions for them. Therefore, the loss values of open-set noisy examples are larger than those of clean examples and lower than those of closed-set noisy examples.

Based on the above observations, in our framework, we employ the powerful pre-trained teacher network to calculate the SL loss [43] for every image example to depict the loss characteristics of different types of data. More specifically, for a possible noisy example $(\mathbf{x}_i, \tilde{y}_i)$ input into the teacher network, the SL loss $\ell_T(\mathbf{x}_i, \tilde{y}_i)$ is calculated as:

$$\ell_T(\mathbf{x}_i, \tilde{y}_i) = \sum_{k=1}^{K} (\tilde{y}_i(k) - \alpha_{ik}/S_i)^2 + \frac{\alpha_{ik}(S_i - \alpha_{ik})}{S_i^2(S_i + 1)}, \quad (2)$$

where $\alpha_{ik} = e_{ik} + 1$ denotes the Dirichlet parameter with e representing the evidence which is the output of the teacher network processed by ReLU activation function, $S_i = \sum_{k=1}^{K} \alpha_{ik}$ means the Dirichlet strength. For the images

with noisy labels, their ground-truth labels are unknown, which introduces classification uncertainty during the neural network training. According to [43], SL loss places a Dirichlet distribution on the model's class probabilities based on Dempster-Shafer Theory of Evidence [3], which enables the model to express the opinion of "I do not know" and overcomes the uncertainty caused by the unknown. Therefore, SL loss can describe the uncertainty caused by noisy labels and capture the discrepancies between the examples of different data types.

To utilize the obtained SL loss values $\ell_T(\mathbf{x}, \tilde{y})$ for data division, we use a Gaussian mixture model (GMM) $\mathcal{G}$ with ψ components $\{g_j\}_{j=1}^{\psi}$ (we set $\psi = 20$) to fit them. Concretely, for $\ell_T(\mathbf{x}, \tilde{y})$, $\mathcal{G}$ is represented as:

$$\mathcal{G}(\ell_T(\mathbf{x}, \tilde{y}) \mid w, \mu, \sigma^2) = \sum_{j=1}^{\psi} w_j g_j \left(\ell_T(\mathbf{x}, \tilde{y}) \mid \mu_j, \sigma_j^2\right), \tag{3}$$

where $w_j > 0$ denotes the weight for the j-th component g_j,[1] that is formulated as:

$$g_j\left(\ell_T(\mathbf{x}, \tilde{y}) \mid \mu_j, \sigma_j^2\right) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(\ell_T(\mathbf{x}, \tilde{y}) - \mu_j)^2}{2\sigma_j^2}\right), \tag{4}$$

where μ_j and σ_j^2 denote the mean and variance of g_j. The weights $\{w_j\}_{j=1}^{\psi}$ satisfy the constraint $\sum_{j=1}^{\psi} w_j = 1$. The parameters $w = \{w_1, w_2, \cdots, w_\psi\}$, $\mu = \{\mu_1, \mu_2, \cdots, \mu_\psi\}$, $\sigma^2 = \{\sigma_1^2, \sigma_2^2, \cdots, \sigma_\psi^2\}$ are estimated by Expectation-Maximization algorithm.

Based on the above analysis, we assume that if a component in GMM is with a small mean, it inclines to capture the possible clean examples. Consequently, we set a minimum threshold $\tau_{min} = 0.3$ to determine the clean examples, and the probability of $\mathbf{x}_i$ belonging to $\mathcal{D}_c$ is $p_i^c = \sum_{j=1, g_j \in \mathcal{G}_c}^{\psi} w_j g_j \left(\ell_T(\mathbf{x}_i, \tilde{y}_i) \mid \mu_j, \sigma_j^2\right)$, where $\mathcal{G}_c$ denotes a set of Gaussian components and each component $g_j \in \mathcal{G}_c$ satisfies $\mu_j \leq \tau_{\text{min}}$. Similarly, we set a maximum threshold $\tau_{max} = 0.9$ and the probability of $\mathbf{x}_i$ belonging to $\tilde{\mathcal{D}}_{cn}$ is $p_i^{cn} = \sum_{j=1, g_j \in \mathcal{G}_{cn}}^{\psi} w_j g_j \left(\ell_T(\mathbf{x}_i, \tilde{y}_i) \mid \mu_j, \sigma_j^2\right)$, where $g_j \in \mathcal{G}_{cn}$ is with mean $\mu_j \geq \tau_{\text{max}}$. Besides, the probability of $\mathbf{x}_i$ belonging to $\tilde{\mathcal{D}}_{on}$ is $p_i^{on} = \sum_{j=1, g_j \in \mathcal{G}_{on}}^{\psi} w_j g_j \left(\ell_T(\mathbf{x}_i, \tilde{y}_i) \mid \mu_j, \sigma_j^2\right)$, where $g_j \in \mathcal{G}_{on}$ is with $\mu_j \in (\tau_{\text{min}}, \tau_{\text{max}})$. Finally, the probabilities of any $(\mathbf{x}_i, \tilde{y}_i) \in \tilde{\mathcal{D}}$ belonging to $\mathcal{D}_c$, $\tilde{\mathcal{D}}_{cn}$, and $\tilde{\mathcal{D}}_{on}$ can be obtained, which are denoted as p_i^c, p_i^{cn}, and p_i^{on}, respectively. Based on $\{p_i^c\}_{i=1}^{|\tilde{\mathcal{D}}|}$, $\{p_i^{cn}\}_{i=1}^{|\tilde{\mathcal{D}}|}$, and $\{p_i^{on}\}_{i=1}^{|\tilde{\mathcal{D}}|}$, we can divide examples in $\tilde{\mathcal{D}}$ into correct sets. For instance, given an image $\mathbf{x}_i$, its division function $f(p_i^c, p_i^{cn}, p_i^{on})$ can be described as:

$$f(p_i^c, p_i^{cn}, p_i^{on}) = \begin{cases} \mathbf{x}_i \in \mathcal{D}_c, & (p_i^c > p_i^{cn}) \& (p_i^c > p_i^{on}), \\ \mathbf{x}_i \in \tilde{\mathcal{D}}_{cn}, & (p_i^{cn} > p_i^c) \& (p_i^{cn} > p_i^{on}), \\ \mathbf{x}_i \in \tilde{\mathcal{D}}_{on}, & (p_i^{on} > p_i^c) \& (p_i^{on} > p_i^{cn}). \end{cases} \tag{5}$$

[1] In this paper, g_j is the shorthand for $g_j\left(\ell_T(\mathbf{x}, \tilde{y}) \mid \mu_j, \sigma_j^2\right)$.

To acquire a small student network, for an image $\mathbf{x}_i \in \mathcal{D}_c$, we retain its label $\tilde{y}_i \in \mathcal{Y}$ as the target y_i for distillation as the label is deemed as correct. For an image $\mathbf{x}_i \in \tilde{\mathcal{D}}_{cn}$ with closed-set noisy label $\tilde{y}_i \in \mathcal{Y}$, the pre-trained $\mathcal{N}_T$ can produce confident outputs of them. Therefore, we use the class label with the largest probability predicted by $\mathcal{N}_T$ as the target y_i, namely:

$$y_i = \arg\max_j \left(\mathcal{N}_T(\mathbf{x}_i)\right)_j. \tag{6}$$

The images in refined $\tilde{\mathcal{D}}_{cn}$ (denoted as $\mathcal{D}_{cn}$) and $\mathcal{D}_c$ with accurate labels are utilized for transferring knowledge from teacher network to student network, which will be detailed in Section IV-B. For the image $\mathbf{x}_i \in \tilde{\mathcal{D}}_{on}$ with unknown true label $y_i \notin \mathcal{Y}$, we discard its annotated noisy label $\tilde{y}_i \in \mathcal{Y}$ and treat them as unlabeled, and depict how to tackle them in Section IV-C.

B. Knowledge Distillation

The key task of knowledge distillation is to transfer the knowledge from teacher network to student network. In the setup of our problem, the teacher network is available. Besides, the clean set $\mathcal{D}_c$ and refined closed noisy set $\mathcal{D}_{cn}$ built in Section IV-A are also at hand for training student network. Therefore, we encourage the student network to learn the knowledge of teacher network in representation space via two aspects: the first is categorical knowledge of images inherited by teacher, and the other is the instance-level representation of teacher. Such learning process is fulfilled by conducting contrastive learning on $\mathcal{D}_c$ and $\mathcal{D}_{cn}$, and the hybrid loss function $\mathcal{L}_{hybrid}$ for transferring knowledge from teacher to student is established as:

$$\mathcal{L}_{hybrid} = \mathcal{L}_{category} + \mathcal{L}_{instance}, \tag{7}$$

where $\mathcal{L}_{category}$ denotes the loss function for promoting student network to learn the categorical knowledge from teacher network, and $\mathcal{L}_{instance}$ represents the loss function for improving the consistency between student network and teacher network in instance level. Next, we will describe the formations of $\mathcal{L}_{category}$ and $\mathcal{L}_{instance}$ in detail.

1) Formation of $\mathcal{L}_{category}$: To encourage student network to learn the categorical knowledge from teacher network, we first calculate the class prototypes by teacher network to reflect its inherited class information. Concretely, we input the images from the same category among the totally K known classes into teacher network to calculate their image embeddings. For class $k \in \{1, 2, \cdots, K\}$, we suppose there are N_k images belonging to this class judged by teacher network. An image $\mathbf{x}_i$ is input into teacher network to obtain the hidden feature $\mathbf{h}_i^T = \mathcal{N}_T(\mathbf{x}_i)$, where $\mathbf{h}_i^T$ is the output after the last average pooling layer of teacher network. Furthermore, we use a projection head that is instantiated as a linear layer with a ℓ_2 normalization to calculate the embedding $\mathbf{z}_i^T$ (*i.e.*, a column vector, see Fig. 2 for image $\mathbf{x}_i$, which is:

$$\mathbf{z}_i^T = \text{Normalization}\left(\mathbf{W}_p^T \mathbf{h}_i^T + \mathbf{b}_p^T\right), \tag{8}$$

where $\mathbf{W}_p^T$ and $\mathbf{b}_p^T$ denote the weights and biases of teacher's projection head. Based on a set of embeddings $\{\mathbf{z}_i^T\}_{i=1}^{N_k}$, the

class prototype $\hat{\mathbf{z}}^k$ of class k can be calculated as:

$$\mathbf{z}^k = \frac{1}{N_k}\sum_{i=1}^{N_k}\mathbf{z}_i^T, \quad \hat{\mathbf{z}}^k = \frac{\mathbf{z}^k}{\left\|\mathbf{z}^k\right\|_2}. \tag{9}$$

Subsequently, we employ these calculated class prototypes $\{\hat{\mathbf{z}}^k\}_{k=1}^K$ to promote student network to learn expressive category-level representation from teacher network.

In a nutshell, we expect that the image embeddings computed by the student network are comparable with those of teacher network. It means that the images in a certain category should have an embedding close to the corresponding class prototype and far away from the prototypes of other classes. Therefore, we use contrastive learning [17], [52] to achieve our target. Contrastive learning devotes to minimizing (maximizing) the distances between positive (negative) pairs. In our case, the images of the same class can be positively paired with their corresponding class prototype and negatively paired with other class prototypes. Therefore, we perform the prototypical contrastive learning to promote the student network to learn the desired representation from teacher network. For an image $\mathbf{x}_i$, similar to the embedding $\mathbf{z}_i^T$ from teacher network mentioned above, we also input it to student network $\mathcal{N}_S$ and its projection head with weights $\mathbf{W}_p^S$ and biases $\mathbf{b}_p^S$ to obtain the embedding $\mathbf{z}_i^S$ (see Fig. 2) from student network, namely:

$$\mathbf{z}_i^S = \text{Normalization}\left(\mathbf{W}_p^S\mathbf{h}_i^S + \mathbf{b}_p^S\right), \tag{10}$$

where the $\mathbf{h}_i^S$ denotes the hidden feature for $\mathbf{x}_i$ that is produced by student network. Taking $\mathbf{z}_i^S$ as an example, the **p**rototypical **c**ontrastive **l**earning loss of student network is calculated as:

$$\mathcal{L}_{pcl}\left(\mathbf{z}_i^S, y_i\right) = -\log\frac{\exp\left(\mathbf{z}_i^S\cdot(\hat{\mathbf{z}}^{y_i})^\top/t\right)}{\sum_{k=1}^K\exp\left(\mathbf{z}_i^S\cdot(\hat{\mathbf{z}}^k)^\top/t\right)}, \tag{11}$$

where $t > 0$ is a temperature parameter, $\hat{\mathbf{z}}^{y_i}$ is the class prototype of class y_i that $\mathbf{x}_i$ belongs to, and the superscript "$\top$" denotes the transpose operation.

Here, $\mathcal{L}_{pcl}\left(\mathbf{z}_i^S, y_i\right)$ depicts the relationship between student's embedding $\mathbf{z}_i^S$ and its corresponding class prototype $\hat{\mathbf{z}}^{y_i}$. Meanwhile, we compute the prototypical contrastive learning loss of teacher network as $\mathcal{L}_{pcl}\left(\mathbf{z}_i^T, y_i\right)$, which represents the relationship between teacher's embedding $\mathbf{z}_i^T$ and its corresponding class prototype $\hat{\mathbf{z}}^{y_i}$. Some recent studies [39], [54], [55] have indicated that transferring the relationships between representations is more effective than actual representations themselves. Therefore, by following [62], we assemble $\mathcal{L}_{pcl}\left(\mathbf{z}_i^T, y_i\right)$ and $\mathcal{L}_{pcl}\left(\mathbf{z}_i^S, y_i\right)$ to transfer the representation relationships from teacher network to student network, and thus boosting student network to learn the categorical knowledge from teacher network. For a batch of image data $\{(\mathbf{x}_i, y_i)\}_{i=1}^b \in (\mathcal{D}_c \cup \mathcal{D}_{cn})$, the loss function $\mathcal{L}_{category}$ for transferring categorical knowledge is defined as:

$$\mathcal{L}_{category} = \frac{1}{b}\sum_{i=1}^b\mathcal{L}_{pcl}\left(\mathbf{z}_i^T, y_i\right) + \mathcal{L}_{pcl}\left(\mathbf{z}_i^S, y_i\right). \tag{12}$$

Note that both $\mathcal{L}_{pcl}\left(\mathbf{z}_i^T, y_i\right)$ and $\mathcal{L}_{pcl}\left(\mathbf{z}_i^S, y_i\right)$ use the class prototypes $\left\{\hat{\mathbf{z}}^k\right\}_{k=1}^K$ rendered by teacher network. During the training, by minimizing $\mathcal{L}_{pcl}\left(\mathbf{z}_i^T, y_i\right)$ and $\mathcal{L}_{pcl}\left(\mathbf{z}_i^S, y_i\right)$, we can update the parameters in teacher's projection head, student's projection head, and student model $\mathcal{N}_S$. Note that, the parameters in teacher model $\mathcal{N}_T$ are fixed.

2) Formation of $\mathcal{L}_{instance}$: Apart from promoting student network to learn categorical knowledge from teacher network, we also encourage student network to learn knowledge from teacher network by considering instance-level consistency. This is also achieved by contrastive learning by following [44], so that the representation capability of student network can be further improved. That is to say, for the same image, we expect student network can produce consistent embedding with teacher network and vice versa. Specifically, for a batch of images $\{(\mathbf{x}_i, y_i)\}_{i=1}^b$, we input them into student network followed by a projection head to obtain the embeddings $\{\mathbf{z}_i^S\}_{i=1}^b$ as Eq. (10), and also input them into teacher network with a projection head to calculate the embeddings $\{\mathbf{z}_i^T\}_{i=1}^b$ as Eq. (8). Formally, for each embedding $\mathbf{z}_i^S$ output by student network, it will form only one positive pair $\{\mathbf{z}_i^S, \mathbf{z}_i^T\}$ and $b-1$ negative pairs $\{\mathbf{z}_i^S, \mathbf{z}_j^T\}_{j=1,i\neq j}^b$. We aim to promote $\mathbf{z}_i^S$ closer to $\mathbf{z}_i^T$ while away from $\{\mathbf{z}_j^T\}_{j=1,i\neq j}^b$. To achieve it, the instance-level contrastive loss $\mathcal{L}_{instance}$ is designed to improve the instance consistency between student network and teacher network, which is:

$$\mathcal{L}_{instance} = \frac{1}{b}\sum_{i=1}^b\log\left(1+\sum_{j=1}^b\mathbb{1}(i\neq j)\exp\left(\mathbf{z}_i^S\cdot(\mathbf{z}_j^T)^\top - \mathbf{z}_i^S\cdot(\mathbf{z}_i^T)^\top\right)\right), \tag{13}$$

where $\mathbb{1}(i\neq j)\in\{0,1\}$ is an indicator function, and its value is 1 if $i\neq j$ and 0 otherwise.

C. Noisy Data Handling

As mentioned in introduction, the main difficulty for our data-free knowledge distillation is to tackle the universal label noise within $\tilde{\mathcal{D}}$ collected from the Internet. In general, after dividing $\tilde{\mathcal{D}}$ into $\mathcal{D}_c$, $\mathcal{D}_{cn}$, and $\tilde{\mathcal{D}}_{on}$ as in Section IV-A, two sources of label noise may appear. The first is that $\mathcal{D}_c$ and $\mathcal{D}_{cn}$ might still contain some label noise due to the imperfect teacher, which will mislead the above categorical knowledge transfer process and lead to undesirable representation learned by student. The second is that there are massive open-set noisy examples in $\tilde{\mathcal{D}}_{on}$. Next, we will detail the strategies for processing the above noise during the training of student network.

1) Tackling Label Noise in $\mathcal{D}_c$ and $\mathcal{D}_{cn}$: For a small number of possible noisy examples in $\mathcal{D}_c$ and $\mathcal{D}_{cn}$, they cannot be completely filtered out as the teacher network may still make incorrect label predictions. Therefore, we use Mixup [59], which has been demonstrated to be noise-robust to deal with label noise. For a batch of images $\{(\mathbf{x}_i, y_i)\}_{i=1}^b$, an image $\mathbf{x}_i$ is linearly interpolated with another randomly selected image $\mathbf{x}_j$ $(i\neq j)$ to produce a virtual training image $\bar{\mathbf{x}}_i$ as:

$$\bar{\mathbf{x}}_i = \lambda\mathbf{x}_i + (1-\lambda)\mathbf{x}_j, \tag{14}$$

where $\lambda \in (0, 1)$ is generated by beta distribution Beta(δ, δ) with δ being a positive parameter. Then, the noise-resistant Mixup on prototypical contrastive learning loss of student network is expressed as:

$$\mathcal{L}_{mixup}\left(\bar{\mathbf{z}}_i^S, y_i, y_j\right) = \lambda \mathcal{L}_{pcl}\left(\bar{\mathbf{z}}_i^S, y_i\right) + (1-\lambda)\mathcal{L}_{pcl}\left(\bar{\mathbf{z}}_i^S, y_j\right), \tag{15}$$

where $\bar{\mathbf{z}}_i^S$ denotes the embedding for $\bar{\mathbf{x}}_i$ that is generated by student network, and $\mathcal{L}_{pcl}$ is presented in Eq. (11). The $\mathcal{L}_{mixup}$ can promote the linearity between the embeddings of the interpolated inputs, therefore enhancing the robustness of the model. Subsequently, the loss function for transferring categorical knowledge from teacher to student in Eq. (12) is rewritten as:

$$\mathcal{L}_{category} = \frac{1}{b}\sum_{i=1}^{b} \mathcal{L}_{mixup}\left(\bar{\mathbf{z}}_i^T, y_i, y_j\right) + \mathcal{L}_{mixup}\left(\bar{\mathbf{z}}_i^S, y_i, y_j\right). \tag{16}$$

2) Tackling Open-set Noisy Examples in $\tilde{\mathcal{D}}_{on}$: Open-set noisy examples do not belong to any classes of the original dataset and are useless for DNN training at a glance. However, there are abundant examples in $\tilde{\mathcal{D}}_{on}$, which contain plentiful information for DNN training. Recent studies [23], [38] in self-supervised learning propose to explore the unlabeled examples to learn a DNN with powerful representational capability, which is useful for downstream tasks. Therefore, we reasonably treat the open-set noisy examples as unlabeled and use them to enhance the representation capability of student network as much as possible.

Concretely, we employ self-supervised learning on rotation recognition to explore the information contained by images in $\tilde{\mathcal{D}}_{on}$. For a batch of unlabeled open-set noisy data $\{\mathbf{x}_i\}_{i=1}^b \in \tilde{\mathcal{D}}_{on}$, each image $\mathbf{x}_i$ is first rotated by $(j-1)\times 90°$ to obtain four counterparts $\mathbf{x}_{i,j}$ $(j \in \{1, 2, 3, 4\})$. Then, these counterparts are subsequently input into student network $\mathcal{N}_S$ and a rotation prediction head with a linear layer to get the rotation prediction $q_{i,j}^S$ (see Fig. 2) as:

$$q_{i,j}^S = \mathbf{W}_r \mathbf{h}_{i,j}^S + \mathbf{b}_r, \tag{17}$$

where $\mathbf{W}_r$ and $\mathbf{b}_r$ denote the weights and biases of the rotation prediction head, and $\mathbf{h}_{i,j}^S$ denotes the hidden feature for $\mathbf{x}_{i,j}$ that is produced by student network. Based on the rotation prediction $q_{i,j}^S$ of image $\mathbf{x}_{i,j}$, the rotation recognition loss $\mathcal{L}_{rr}$ to exploit massive open-set noisy examples for student model is calculated as:

$$\mathcal{L}_{rr} = \frac{1}{4b}\sum_{i=1}^{b}\sum_{j=1}^{4} \mathcal{H}_{ce}\left(q_{i,j}^S, j\right), \tag{18}$$

where $\mathcal{H}_{ce}$ is the cross entropy loss to compute the discrepancy between the rotation prediction $q_{i,j}^S$ and its rotated angle $(j-1)\times 90°$.

Moreover, like our method, some prior works [20], [21], [27], [47] on label noise learning and open-set semi-supervised learning also propose different ways to utilize noisy or open-set data. For example, Dividemix [27] and ProMix [47] treat the noisy examples as unlabeled and utilize them to improve the model training by semi-supervised learning. Trash to Treasure [20] and Transferable OOD Data Recycling [21] extract additional supervisory signals from open-set examples via contrastive learning or adversarial domain adaptation. That is to say, our approach is well grounded given previous similar works. The main difference lies in that we utilize plentiful open-set noisy examples via self-learning task on rotation recognition to further improve the representation ability of our student network.

D. Overall Loss Function & Algorithm Implementation

This section introduces the complete loss function employed by our MODUL and also provides the details for implementing our algorithm. To enable student network to make accurate classification, for a batch of data $\{(\mathbf{x}_i, y_i)\}_{i=1}^b \in (\mathcal{D}_c \cup \mathcal{D}_{cn})$, we follow [18] and use cross entropy loss to penalize the difference between the output of student and the label y in $\mathcal{D}_c$ and $\mathcal{D}_{cn}$, namely:

$$\mathcal{L}_{ce} = \frac{1}{b}\sum_{i=1}^{b} \mathcal{H}_{ce}\left(\mathcal{N}_S(\mathbf{x}_i), y_i\right). \tag{19}$$

To promote the student network to learn the knowledge from teacher network, we also use $\mathcal{D}_c$ and $\mathcal{D}_{cn}$ to transfer the meaningful information of teacher to student by the widely used knowledge distillation loss [41], which is:

$$\mathcal{L}_{kd} = \frac{1}{b}\sum_{i=1}^{b} \mathcal{H}_{mse}\left(\mathbf{f}_i^S, \mathbf{f}_i^T\right), \tag{20}$$

where $\mathcal{H}_{mse}$ denotes the mean-square error loss, $\mathbf{f}_i^T$ and $\mathbf{f}_i^S$ represent the penultimate layer's features of teacher network and student network, respectively. Then, the complete objective loss function of our MODUL can be formulated as:

$$\mathcal{L}_{objective} = \alpha \mathcal{L}_{ce} + (1-\alpha)\mathcal{L}_{kd} + \beta \mathcal{L}_{hybrid} + \gamma \mathcal{L}_{rr}, \tag{21}$$

where α, β, and γ are non-negative trade-off parameters.

The training algorithm of our MODUL is summarized in Alg. 1, which contains two stages. In Stage 1, all examples in $\tilde{\mathcal{D}}$ are attributed into $\mathcal{D}_c$, $\mathcal{D}_{cn}$, and $\tilde{\mathcal{D}}_{on}$. In Stage 2, all examples in $\mathcal{D}_c$, $\mathcal{D}_{cn}$, and $\tilde{\mathcal{D}}_{on}$ are effectively explored, and the trained student network not only possesses informative knowledge but also is robust to universal label noise. Here the cross entropy loss $\mathcal{L}_{ce}$ in the overall loss function $\mathcal{L}_{objective}$ can be substituted by some specialized loss functions [9], [13], [60] tailored for handling noisy labels to further mitigate the potential noisy labels. However, here we simply employ the common cross entropy loss because our method has already achieved good noise-correction performance due to the well-trained teacher network and the robust Mixup operation. Therefore, for the sake of simplicity in algorithm implementation, we choose to use conventional cross entropy loss in our method.

V. Experiments

This section describes the noisy dataset construction (Section V-A), verifies the effectiveness of our MODUL under

Algorithm 1 MOdel Distillation With Universal Label Noise

Input: A large pre-trained teacher network $\mathcal{N}_T$, noisy dataset $\tilde{\mathcal{D}} = \{(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^{|\tilde{\mathcal{D}}|}$, threshold parameters τ_{min} and τ_{max}, trade-off parameters α, β, and γ.

Stage 1: Noisy data division.
Calculate SL loss values $\{\ell_T(\mathbf{x}_i, \tilde{y}_i)\}_{i=1}^{|\tilde{\mathcal{D}}|}$ via Eq. (2) and use GMM to fit them;
Calculate the probabilities of images belonging to $\tilde{\mathcal{D}}_c$, $\tilde{\mathcal{D}}_{cn}$, and $\tilde{\mathcal{D}}_{on}$, which are $\{p_i^c\}_{i=1}^{|\tilde{\mathcal{D}}|}$, $\{p_i^{cn}\}_{i=1}^{|\tilde{\mathcal{D}}|}$, and $\{p_i^{on}\}_{i=1}^{|\tilde{\mathcal{D}}|}$, respectively;
Attribute $\mathbf{x}_i$ to $\tilde{\mathcal{D}}_c$, $\tilde{\mathcal{D}}_{cn}$ or $\tilde{\mathcal{D}}_{on}$ via Eq. (5);
Refine labels in $\tilde{\mathcal{D}}_{cn}$ via Eq. (6);
Stage 2: Learning student network $\mathcal{N}_S$.
Initialize the small student network $\mathcal{N}_S$;
Compute the category prototypes $\{\hat{\mathbf{z}}^k\}_{k=1}^K$ via Eq. (9);
repeat
  Randomly select $\{(\mathbf{x}_i, y_i)\}_{i=1}^b \in (\tilde{\mathcal{D}}_c \cup \tilde{\mathcal{D}}_{cn})$;
  Randomly select $\{\mathbf{x}_i\}_{i=1}^b \in \tilde{\mathcal{D}}_{on}$;
  Produce the virtual image example $\bar{\mathbf{x}}_i$ via Eq. (14);
  Calculate categorical knowledge transfer loss $\mathcal{L}_{category}$ via Eq. (16);
  Calculate instance-level contrastive loss $\mathcal{L}_{instance}$ via Eq. (13);
  Conduct image augmentation by rotating $\mathbf{x}_i$ to $\{\mathbf{x}_{i,j}\}_{j=1}^4$;
  Calculate rotation recognition loss $\mathcal{L}_{rr}$ via Eq. (18);
  Calculate total objective loss $\mathcal{L}_{objective}$ via Eq. (21);
  Update student network $\mathcal{N}_S$ via SGD;
until convergence

Output: Small student network $\mathcal{N}_S$.

different noise levels (Section V-B), compares the proposed MODUL with other data-free knowledge distillation methods (Section V-C), justifies the usefulness of key operations in MODUL (Section V-E), and analyzes the parametric sensitivity of the pre-tuned parameters (Section V-F).

A. Noisy Dataset Construction

In this section, we introduce the way for building noisy datasets under universal label noise for our empirical study. By following prior works [42] on label noise learning, we utilize CIFAR-10 and CIFAR-100 [25] datasets as in-distribution data, and extract out-of-distribution data from large-scale ImageNet [10] to mimic the massive examples collected from Internet. CIFAR-10 (CIFAR-100) consists of 60,000 32×32 RGB images of 10 (100) categories, among which 50,000 images are for training, and 10,000 images are for testing. ImageNet contains up to 1,000 categories and about 1.2 million images, far exceeding those of CIFAR-10 and CIFAR-100.

Here, we construct two noisy datasets, namely "CIFAR-10 + ImageNet" and "CIFAR-100 + ImageNet". Specifically, the constructed noisy datasets are governed by two noise ratios ρ_1 and ρ_2 introduced in Section IV-A, where ρ_1 denotes the ratio of noisy examples in the entire dataset, and ρ_2 means the proportion of open-set noisy examples

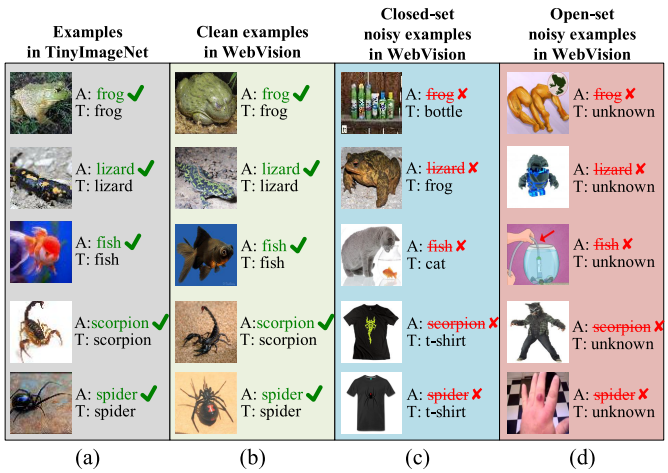


Fig. 4. Visualization of examples in TinyImageNet and WebVision datasets. (a) shows the examples in TinyImageNet, which are used to train teacher network. For the training images of student network in the realistic WebVision, (b) shows the clean examples that are annotated with correct true labels; (c) exhibits the closed-set noisy examples that are incorrectly labeled as other known classes-of-interests; and (d) displays the open-set noisy examples with unknown true labels.

in all noisy examples. For the noisy dataset CIFAR-10 (CIFAR-100) + ImageNet contains 50,000 training examples, which can be divided into clean set, closed noisy set, and open noisy set. For clean set, we randomly select $(1-\rho_1) \times 50,000$ images from CIFAR-10 (CIFAR-100) and retain their original labels. For closed noisy set, we randomly choose $\rho_1 \times (1-\rho_2) \times 50,000$ images from CIFAR-10 (CIFAR-100) and flip their labels to another class that belongs to CIFAR-10 (CIFAR-100). For open noisy set, we randomly extract $\rho_1 \times \rho_2 \times 50,000$ images from ImageNet and assign a class label within CIFAR-10 (CIFAR-100) to them. Same as [18], we perform random flip, random crop, and zero padding on all selected images for data augmentation. To make the image size consistent, we down-sampled the images in ImageNet to the size of 32×32. The CIFAR and ImageNet datasets contain some overlapped categories. Therefore, we removed the images in ImageNet belonging to these overlapped categories to avoid selecting them as out-of-distribution data.

Additionally, we follow recent studies [7], [27] in semi-supervised learning to train student network on a realistic noisy dataset composed of the images in the wild. That is to say, we use TinyImageNet [26] as the original data for teacher network, and train student network on the corresponding images in WebVision [30], which belongs to the 200 categories in TinyImageNet and is acquired according to the data provided by the official website of WebVision.[2] TinyImageNet is a popular subset of ImageNet dataset, which contains 64×64 RGB images of 200 categories. Each class has 500 training images, 50 validation images, and 50 test images. WebVision composed of over 2.4 million unlabeled images crawled from the Internet using the 1000 categories in ImageNet as indices. As shown in Fig. 4, in the images of WebVision, there are many closed-set and open-set noisy examples. Therefore, training student network on WebVision

[2]https://data.vision.ee.ethz.ch/cvl/webvision/dataset2017.html

TABLE I

CLASSIFICATION RESULTS OF THE EXPERIMENTS ON CIFAR-10 + IMAGENET AND CIFAR-100 + IMAGENET, WHERE THE NOISE RATIOS $\rho_1, \rho_2 \in \{0.25, 0.50, 0.75\}$

Noisy dataset	Algorithm	$\rho_1 = 0.25$			$\rho_1 = 0.50$			$\rho_1 = 0.75$		
		$\rho_2 = 0.25$	$\rho_2 = 0.50$	$\rho_2 = 0.75$	$\rho_2 = 0.25$	$\rho_2 = 0.50$	$\rho_2 = 0.75$	$\rho_2 = 0.25$	$\rho_2 = 0.50$	$\rho_2 = 0.75$
CIFAR-10 + ImageNet	VKD [19]	88.46%	88.35%	88.11%	87.90%	87.53%	86.37%	87.68%	86.64%	85.78%
	DFND [5]	94.29%	94.01%	93.54%	93.75%	93.41%	92.81%	93.37%	93.09%	92.17%
	MODUL	**96.27%**	**96.13%**	**95.72%**	**96.15%**	**95.81%**	**95.30%**	**95.35%**	**95.26%**	**94.16%**
CIFAR-100 + ImageNet	VKD [19]	64.61%	64.15%	63.81%	64.39%	62.84%	61.79%	64.04%	61.13%	59.22%
	DFND [5]	77.19%	76.74%	76.03%	76.48%	75.40%	72.73%	74.21%	72.83%	66.26%
	MODUL	**79.53%**	**78.95%**	**78.15%**	**78.93%**	**78.37%**	**77.30%**	**77.16%**	**75.40%**	**73.55%**

can faithfully evaluate the effectiveness of our method in solving universal label noise.

B. Experiments on Various Noisy Datasets

In this section, we conduct extensive experiments on the constructed noisy datasets "CIFAR-10 + ImageNet" and "CIFAR-100 + ImageNet" under various noise ratios $\rho_1, \rho_2 \in \{0.25, 0.50, 0.75\}$, and the realistic noisy dataset WebVision to evaluate the effectiveness of our MODUL. To our best knowledge, Data-Free Noisy Distillation (DFND) [5] is the only existing data-free model compression method that can utilize noisy data, which processes the closed-set label noise by a noise adaption matrix. Therefore, we compare our MODUL with DFND to demonstrate the capability of our MODUL in handling noisy data. Meanwhile, we train a student network on the noisy datasets by Vanilla Knowledge Distillation (VKD) [19] to show the necessity of label noise processing, where the label noise in dataset is ignored.

To achieve fair competition, all the compared methods share the same backbone networks. We utilize two popular teacher-student pairs ResNet34-ResNet18 and VGGNet16-VGGNet13. During training, we select Stochastic Gradient Descent (SGD) as optimizer and set the weight decay and momentum parameters as 10^{-4} and 0.9, respectively. We train all networks for 200 epochs with batch-size $b = 64$, and the learning rate is initially set as 0.1 and divided by ten at 80 and 160 epochs. Besides, the trade-off parameters in Eq. (21) are set to $\alpha = \beta = 0.1$ and $\gamma = 0.01$, and the temperature parameter t in Eq. (11) is set to 0.3. The parametric sensitivity will be studied in Section V-F.

Table I reports the classification results on the noisy datasets CIFAR-10/100 + ImageNet and TinyImageNet + WebVision. Firstly, we observe that noisy labels will damage the effect of knowledge distillation. VKD neglects label noise and performs poorly on noisy datasets over various noise levels. In contrast, DFND and our MODUL can handle noisy labels, so their performance is significantly better than VKD. Secondly, we can see that our MODUL consistently outperforms DFND on noisy datasets over various noise levels with a large margin. This is because DFND ignores massive open-set noisy examples that are ubiquitous in the wild, which are still useful for student network training if explored appropriately. In contrast, our MODUL properly utilizes both closed-set and open-set noisy examples. When $\rho_1 = \rho_2 = 0.75$, the accuracy of MODUL is 1.99% and 7.29% higher than that of DFND on CIFAR-10 + ImageNet and CIFAR-100 + ImageNet, respectively. The experimental results demonstrate that the proposed MODUL can effectively process the noisy dataset containing universal label noise and train a compact student network to solve the data-free model compression problem.

TABLE II

CLASSIFICATION RESULTS OF THE STUDENT NETWORKS TRAINED ON THE REALISTIC NOISY DATASET CONSTRUCTED BY THE IMAGES IN THE WILD. THE ORIGINAL DATASET OF TEACHER NETWORK AND NOISY DATASET OF STUDENT NETWORK ARE TINYIMAGENET AND WEBVISION, RESPECTIVELY."RESNET34 (65.75%)" REPRESENTS THAT THE ACCURACY OF RESNET34 TRAINED ON THE ORIGINAL DATASET IS 65.75%, AND THE SAME APPLIES TO OTHER ITEMS

Teacher	Student	VKD [19]	DFND [5]	MODUL
ResNet34 (65.75%)	ResNet18 (64.33%)	48.80%	55.20%	**64.43%**
VGGNet16 (61.58%)	VGGNet13 (60.44%)	46.18%	52.58%	**60.28%**

Table II reports the classification results on the realistic noisy dataset WebVision. We can observe that the student networks trained by our method still achieve comparable performance with the same one trained on the original dataset. For the teacher-student pair ResNet34-ResNet18, the accuracy of our method is 15.63% and 9.23% higher than that of VKD and DFND, respectively. For the teacher-student pair VGGNet16-VGGNet13, the accuracy of our method is 14.10% and 7.70% higher than that of VKD and DFND, respectively. The experimental results demonstrate that the proposed MODUL can effectively process the real-world noisy dataset containing both closed-set label noise and open-set label noise, therefore training a compact and reliable student network.

C. Comparison With Other Data-Free Knowledge Distillation Methods

In this paper, we focus on data-free model compression. Therefore, we further compare the proposed MODUL with other data-free methods, which usually generate fake data from a pre-trained teacher network for training student network, such as:

- Data-Free Learning (DFL) [6], which introduces a generator to synthesize fake data under the supervision of a pre-trained teacher network.
- Data-Free Adversarial Learning (DFAD) [12], which utilizes teacher network and student network to jointly supervise generator to produce fake data.
- DeepInversion [56], which inverts the means and variances stored in teacher's features to generate training images.

TABLE III

CLASSIFICATION RESULTS OF THE PROPOSED MODUL AND OTHER COMPARED METHODS. THE REPORTED ACCURACIES ARE EVALUATED ON THE TEST SETS OF CIFAR-10 AND CIFAR-100 DATASETS. "−" MEANS THAT THE CORRESPONDING METHOD DOES NOT NEED INPUT DATA

Algorithm	Required data	CIFAR-10	CIFAR-100
VKD [19]	Original clean data	94.34%	76.87%
DFL [6]	-	92.22%	74.47%
DFAD [12]	-	93.30%	67.70%
DeepInversion [56]	-	93.26%	61.32%
ZSKT [37]	-	93.32%	67.74%
DFND [5]	Collected noisy data	93.41%	75.40%
MODUL	Collected noisy data	**95.81**%	**78.37**%

- Zero-Shot Knowledge Transfer (ZSKT) [37], which promotes student network to mimic the predictions of teacher network for "hard" examples that are produced by generator.

In addition, we also introduce VKD and DFND appeared in Section V-B for comparison. In this section, VKD is trained on the clean CIFAR-10 and CIFAR-100 datasets, while DFND and our MODUL are trained on CIFAR-10 + ImageNet and CIFAR-100 + ImageNet with $\rho_1 = \rho_2 = 0.50$. The student networks trained by all methods are evaluated on the test sets of CIFAR-10 and CIFAR-100.

The experimental settings of MODUL are the same as those declared in Section V-B. It is worth noting that, for fair a comparison, all baseline methods and our MODUL employ ResNet34 and ResNet18 as teacher network and student network, respectively. The parameters α and β in DFL [6] to balance activation loss and entropy loss for generator are set to 0.01 and 20, respectively. In DFAD [12], the generator is updated one time when the student network is updated every five times. In DeepInversion [56], the parameter α_c for image diversity competition loss is adjusted to 10. Besides, the gradient steps of student and generator during each iteration in ZSKT [37] are set as 10 and 1, respectively. Moreover, the temperature parameter τ of DFND [5] is determined as 4.

Table III presents the classification results. We see that DFL, DFAD, DeepInversion, and ZSKT can generate fake data for student network training, but their performance is poor due to the discrepancy between the generated data and true data. Meanwhile, the performance of DFND is still lower than VKD since it ignores open-set label noise in the collected examples. In particular, our MODUL achieves the best classification performance among all methods. Specifically, the accuracy of the proposed MODUL is 1.47% and 1.50% higher than the same network trained by VKD on CIFAR-10 and CIFAR-100 datasets, respectively. It indicates that even when half of the examples in the dataset are noisy, our MODUL can still acquire a student network to achieve comparable performance with the same network trained on the original clean dataset.

D. Comparison With Other Learning Methods With Noisy Labels

In our approach, we aim to train a compact and reliable student network on the noisy dataset with universal label noise. To achieve this target, it is vital to properly solve the potential noisy data in the divided clean set, closed noisy set, and open noisy set. Therefore, we compare our method with state-of-the-art methods of learning with noisy labels to evaluate the effectiveness of our method for combating label noise, including:

- Reweighting [32], which assigns different weights to noisy examples based on their estimated probability of being noisy.
- Dimensionality-Driven Learning (D2L) [35], which modifies the loss of DNNs at their dimensionality expansion stage that is usually prone to noisy labels.
- Generalized Cross Entropy (GCE) [60], which encourages DNNs to learn both the true labels and predicted probabilities.
- Symmetric Cross Entropy (SCE) [50], which can simultaneously solve overfitting on noisy labels in easy categories and underfitting on hard categories.
- Classification Loss with Entropic Optimal Transport (CLEOT) [9], which utilizes entropic optimal transportation to prevent DNNs from overfitting to noisy labels.
- Early-Learning Regularization (ELR) [31], which corrects the noisy labels in the early training stage of DNNs to avoid them remembering these noisy labels.
- Wasserstein Adversarial Regularization (WAR) [13], which transforms the predicted distribution of labels into the true distribution based on Wasserstein distance.

To compare these label noise processing methods with our method, we use them to replace the cross entropy loss in our method and discard hybrid contrastive learning loss $\mathcal{L}_{hybrid}$ (Eq. (7)) and rotation recognition loss $\mathcal{L}_{rr}$ (Eq. (18)). For a fair comparison, all compared methods and our MODUL employ the teacher-student pair ResNet34-ResNet18 to train on CIFAR-10 + ImageNet and CIFAR-100 + ImageNet datasets with $\rho_1 = \rho_2 = 0.50$. The experimental setups are the same as those declared in Section V-B.

TABLE IV

CLASSIFICATION RESULTS OF THE PROPOSED MODUL AND OTHER POPULAR METHODS OF LEARNING WITH NOISY LABELS. ALL STUDENT NETWORKS TRAINED ON NOISY DATASET ARE EVALUATED ON THE TEST SETS OF CIFAR-10 AND CIFAR-100. THE COLUMN "WITHOUT/WITH MIXUP" PRESENTS THE RESULTS PRODUCED BY THE CORRESPONDING METHODS WITHOUT/WITH MIXUP OPERATION

Algorithm	Without Mixup		With Mixup	
	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Cross entropy	87.53%	62.84%	93.55%	75.18%
Reweight [32]	92.16%	73.56%	94.23%	75.22%
D2L [35]	92.23%	72.38%	94.62%	75.40%
GCE [60]	**94.11**%	73.53%	94.91%	76.01%
SCE [50]	93.71%	72.61%	94.45%	75.35%
CLEOT [9]	92.77%	73.62%	94.14%	75.55%
ELR [31]	93.94%	74.32%	94.37%	76.12%
WAR [13]	93.83%	74.07%	94.94%	76.00%
MODUL	93.71%	**76.65**%	**95.81**%	**78.37**%

The experimental results, as reported in Table IV, highlight two main observations. Firstly, the methods involving label noise handling achieve better performance than those only using cross entropy loss function, which demonstrates the necessity of handling potentially noisy examples. Secondly,

TABLE V

CLASSIFICATION RESULTS OF THE ABLATION EXPERIMENTS. THE PERFORMANCE DROP OF EACH SETTING COMPARED WITH COMPLETE MODUL IS INDICATED IN RED FONT IN THE BRACKET

Noisy dataset	Algorithm	$\rho_2 = 0.25$	$\rho_2 = 0.50$	$\rho_2 = 0.75$
CIFAR-10 + ImageNet	No Mixup	94.19% (↓ 1.96%)	93.71% (↓ 2.10%)	93.10% (↓ 2.20%)
	No $\mathcal{L}_{rr}$	95.58% (↓ 0.57%)	95.29% (↓ 0.52%)	94.27% (↓ 1.03%)
	No $\mathcal{L}_{category}$	94.72% (↓ 1.43%)	94.12% (↓ 1.69%)	93.78% (↓ 1.52%)
	No $\mathcal{L}_{instance}$	95.63% (↓ 0.52%)	95.31% (↓ 0.50%)	94.38% (↓ 0.92%)
	No $\mathcal{L}_{hybrid}$	94.04% (↓ 2.11%)	93.76% (↓ 2.05%)	92.88% (↓ 2.42%)
	MODUL	**96.15**%	**95.81**%	**95.30**%
CIFAR-100 + ImageNet	No Mixup	77.44% (↓ 1.49%)	76.65% (↓ 1.72%)	75.07% (↓ 2.23%)
	No $\mathcal{L}_{rr}$	78.49% (↓ 0.44%)	77.40% (↓ 0.97%)	75.84% (↓ 1.46%)
	No $\mathcal{L}_{category}$	78.26% (↓ 0.67%)	77.27% (↓ 1.10%)	75.92% (↓ 1.38%)
	No $\mathcal{L}_{instance}$	78.69% (↓ 0.24%)	77.94% (↓ 0.43%)	76.33% (↓ 0.97%)
	No $\mathcal{L}_{hybrid}$	77.88% (↓ 1.05%)	76.98% (↓ 1.39%)	75.69% (↓ 1.61%)
	MODUL	**78.93**%	**78.37**%	**77.30**%

our method with Mixup performs better than those methods with advanced techniques of learning with noisy labels, which reflects that our method can further effectively address the possible noisy labels in the divided dataset.

E. Verification of Key Operations in MODUL

To further demonstrate the effectiveness of the proposed MODUL, we conduct ablation studies and visualize the divided examples to show that the key operations designed in MODUL are indeed necessary.

1) Ablation Study: To understand the function of each component in the proposed MODUL, we use CIFAR-10 + ImageNet and CIFAR-100 + ImageNet datasets with noise ratios $\rho_1 = 0.50$ and $\rho_2 \in \{0.25, 0.50, 0.75\}$ for our ablative experiments. Specifically, we study the performances of three key components of MODUL by removing the related operations, namely: 1) Open-set noisy examples utilization in Section IV-C (see "No $\mathcal{L}_{rr}$"); 2) Knowledge transfer in Section IV-B (see "No $\mathcal{L}_{hybrid}$", "No $\mathcal{L}_{instance}$", and "No $\mathcal{L}_{category}$"); and 3) Mixup in Section IV-C (see "No Mixup"). The experimental setups are the same as those in Section V-B. Table V shows the experimental results of ablation studies and below we will analyze the contribution of each component in our MODUL:

1) Open-set noisy examples utilization. The performance of student network consistently degrades as the number of open-set noisy examples increases, *i.e.*, ρ_2 rises. The results indicate that our MODUL can effectively utilize the valuable open-set noisy examples.
2) Knowledge transfer. The student network gets poor classification results when the term $\mathcal{L}_{hybrid}$ is removed, as in this case, the student will not learn any categorical or instance knowledge from teacher network. In contrast, as shown in "No $\mathcal{L}_{instance}$" and "No $\mathcal{L}_{category}$", the student network learning category-level knowledge or instance-level knowledge all exhibit considerable performance improvements. In particular, the student network achieves the best performance (shown in "MODUL") when both categorical and instance knowledge are transferred from teacher to student. The results demonstrate that both categorical and instance knowledge are beneficial for student network training.

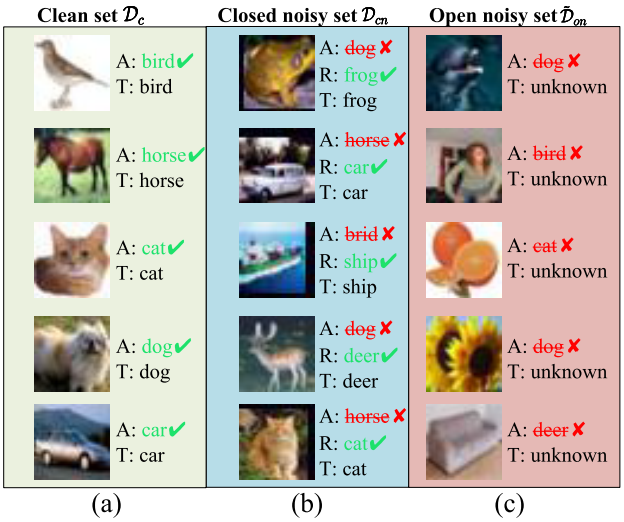
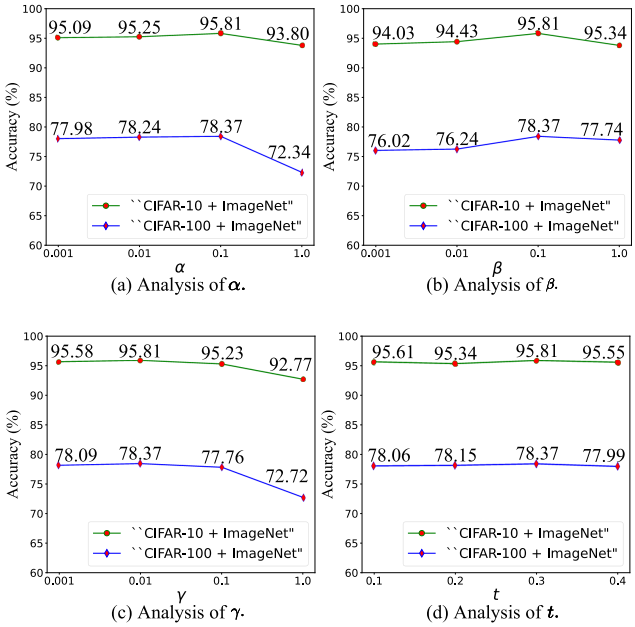


Fig. 5. Visualization of divided examples of the noisy dataset CIFAR-10 + ImageNet with $\rho_1 = \rho_2 = 0.50$. (a) shows that the clean examples are annotated with correct true labels. (b) shows that the labels of closed-set noisy examples can be correctly refined by our method. (c) displays that the open-set noisy examples with unknown true labels can be accurately identified by our method.

3) Mixup. The performance of student network decreased significantly on CIFAR-10 + ImageNet and CIFAR-100 + ImageNet over noisy datasets with different noise levels. The results indicate that Mixup is critical to handling unexpected noisy labels.

2) Visualization of Dataset Division: To show the effectiveness of our proposed dataset division method in Section IV-A, we visualize the examples divided by our method on the noisy dataset CIFAR-10 + ImageNet with $\rho_1 = \rho_2 = 0.50$. Fig. 5 shows the determined images in clean set $\mathcal{D}_c$, closed noisy set $\mathcal{D}_{cn}$, and open noisy set $\tilde{\mathcal{D}}_{on}$, respectively. Specifically, the clean examples in Fig. 5(a) and closed-set noisy examples in Fig. 5(b) contain the images of "cat", "dog", and "ship", etc., which belong to the classes-of-interests within CIFAR-10 dataset. From Fig. 5, we see that the labels of clean examples are retained by our method, as they are consistent with their actual labels. Meanwhile, the labels of mislabeled closed-set noisy examples are precisely refined by our teacher network. Furthermore, the open-set noisy examples in Fig. 5(c) are out-of-distribution data not contained by the CIFAR-10 dataset.



(a) Analysis of α. (b) Analysis of β. (c) Analysis of γ. (d) Analysis of t.

Fig. 6. Parametric sensitivity of (a) α, (b) β, (c) γ, and (d) t in our MODUL on noisy datasets CIFAR-10 + ImageNet and CIFAR-100 + ImageNet with noise ratios $\rho_1 = \rho_2 = 0.50$.

The visualization result demonstrates that the proposed dataset division scheme in Section IV-A can effectively classify the examples in the noisy dataset into $\mathcal{D}_c$, $\mathcal{D}_{cn}$, and $\tilde{\mathcal{D}}_{on}$. Therefore, the student network supervised by teacher network can be well-trained on these divided data.

F. Parametric Sensitivity

Three trade-off parameters α, β, γ, and the temperature parameter t in our MODUL are required to be pre-tuned manually. This section studies the sensitivity of our MODUL to these parameters on the noisy datasets CIFAR-10 + ImageNet and CIFAR-100 + ImageNet with noise ratios $\rho_1 = \rho_2 = 0.50$. The model configurations and experimental setups are the same as those in Section V-B. During training, we examine the produced accuracy by changing one of the four parameters and fixing every remaining parameter to a constant value listed in Section V-B.

Fig. 6 shows the curves of test accuracy produced by the student network when the parameters vary. We can obverse that these parameters cover a wide range, where α, β, and γ are within $\{0.001, 0.01, 0.1, 1.0\}$, and $t \in \{0.1, 0.2, 0.3, 0.4\}$. It can be observed that the curves of accuracies are generally stable, indicating that the performance of student network is robust to the variations of parameters. Therefore, the parameters in our MODUL are easy to tune.

VI. Conclusion and Future Work

This paper proposes a new data-free knowledge distillation framework termed "**MO**del **D**istillation with **U**niversal **L**abel noise" (MODUL) to process the absence of original data. The key of MODUL is to train a compact student network on a webly-collected dataset under universal label noise. Specifically, our MODUL has the following key operations: firstly, it carefully explores the loss value distribution of different data types and precisely divides the noisy data into clean set, closed noisy set, and open noisy set; secondly, student network can learn both category-level and instance-level knowledge from teacher network, and thus exhibiting satisfactory representation ability; and thirdly, it properly tackles the closed-set and open-set label noise by Mixup and self-supervised learning, respectively, leading to a noise-robust student network. The results evaluated on the noisy benchmark datasets indicate that our MODUL outperforms state-of-the-art data-free knowledge distillation methods.

Generally speaking, a stronger teacher network might be more powerful in dealing with label noise. However, in our investigated knowledge distillation scenario, a significant gap between teacher network and student network will also reduce the distillation effect, ultimately decreasing the performance of student network (see Supplementary Materials). Therefore, in future work, we plan to explore the trade-off between the capability of teacher network and the performance gap between teacher and student.

Acknowledgment

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, or the Ministry of Communications and Information.

References

[1] C. An, Y. Wang, J. Zhang, and T. Q. Nguyen, "Self-supervised rigid registration for multimodal retinal images," *IEEE Trans. Image Process.*, vol. 31, pp. 5733–5747, 2022.

[2] J. Ba and R. Caruana, "Do deep nets really need to be deep?" in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 27, 2014.

[3] J. A. Barnett, "Computational methods for a mathematical theory of evidence," in *Classic Works of the Dempster-Shafer Theory of Belief Functions*. Cham, Switzerland: Springer, 2008, pp. 197–216.

[4] D. Chen et al., "Cross-layer distillation with semantic calibration," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 8, pp. 7028–7036.

[5] H. Chen et al., "Learning student networks in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6428–6437.

[6] H. Chen et al., "Data-free learning of student networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3514–3522.

[7] J. Chen et al., "Label-retrieval-augmented diffusion models for learning from noisy labels," 2023, *arXiv:2305.19518*.

[8] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 1597–1607.

[9] B. B. Damodaran, R. Flamary, V. Seguy, and N. Courty, "An entropic optimal transport loss for learning deep neural networks under label noise in remote sensing images," *Comput. Vis. Image Understand.*, vol. 191, Feb. 2020, Art. no. 102863.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[11] G. Ding, S. Zhang, Z. Jia, J. Zhong, and J. Han, "Where to prune: Using LSTM to guide data-dependent soft pruning," *IEEE Trans. Image Process.*, vol. 30, pp. 293–304, 2021.

[12] G. Fang, J. Song, C. Shen, X. Wang, D. Chen, and M. Song, "Data-free adversarial distillation," 2019, *arXiv:1912.11006*.

[13] K. Fatras, B. B. Damodaran, S. Lobry, R. Flamary, D. Tuia, and N. Courty, "Wasserstein adversarial regularization for learning with label noise," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 7296–7306, Oct. 2022.

[14] L. Feng, S. Shu, Z. Lin, F. Lv, L. Li, and B. An, "Can cross entropy loss be robust to label noise?" in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 2206–2212.

[15] A. Ghosh, H. Kumar, and P. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 31, no. 1, 2017, pp. 1919–1925.

[16] B. Han et al., "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. NIPS*, 2018, pp. 8536–8546.

[17] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[19] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.

[20] J. Huang et al., "Trash to treasure: Harvesting OOD data with cross-modal matching for open-set semi-supervised learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8290–8299.

[21] Z. Huang, J. Yang, and C. Gong, "They are not completely useless: Towards recycling transferable unlabeled data for class-mismatched semi-supervised learning," *IEEE Trans. Multimedia*, vol. 25, pp. 1844–1857, 2022.

[22] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2304–2313.

[23] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," in *Proc. IEEE Trans. Pattern Anal. Mach. Intell.*, Nov. 2020, vol. 43, no. 11, pp. 4037–4058.

[24] P. Khosla et al., "Supervised contrastive learning," in *Proc. NIPS*, 2020, pp. 18661–18673.

[25] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.

[26] Y. Le and X. Yang, "Tiny ImageNet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[27] J. Li, R. Socher, and S. C. H. Hoi, "DivideMix: Learning with noisy labels as semi-supervised learning," 2020, *arXiv:2002.07394*.

[28] J. Li, C. Xiong, and S. C. H. Hoi, "Learning from noisy data with robust representation learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9465–9474.

[29] M. Li, C. Li, and J. Guo, "Cluster-guided asymmetric contrastive learning for unsupervised person re-identification," *IEEE Trans. Image Process.*, vol. 31, pp. 3606–3617, 2022.

[30] W. Li, L. Wang, W. Li, E. Agustsson, and L. Van Gool, "WebVision database: Visual learning and understanding from web data," 2017, *arXiv:1708.02862*.

[31] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, "Early-learning regularization prevents memorization of noisy labels," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 33, 2020, pp. 20331–20342.

[32] T. Liu and D. Tao, "Classification with noisy labels by importance reweighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 447–461, Mar. 2016.

[33] Z. Liu, Y. Wang, K. Han, S. Ma, and W. Gao, "Instance-aware dynamic neural network quantization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12434–12443.

[34] R. G. Lopes, S. Fenu, and T. Starner, "Data-free knowledge distillation for deep neural networks," 2017, *arXiv:1710.07535*.

[35] X. Ma et al., "Dimensionality-driven learning with noisy labels," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3355–3364.

[36] P. C. Madhusudana, N. Birkbeck, Y. Wang, B. Adsumilli, and A. C. Bovik, "Image quality assessment using contrastive learning," *IEEE Trans. Image Process.*, vol. 31, pp. 4149–4161, 2022.

[37] P. Micaelli and A. J. Storkey, "Zero-shot knowledge transfer via adversarial belief matching," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019, pp. 9547–9557.

[38] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2016, pp. 69–84.

[39] W. Park, D. Kim, Y. Lu, and M. Cho, "Relational knowledge distillation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3967–3976.

[40] G. Patrini, A. Rozza, A. K. Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1944–1952.

[41] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*.

[42] R. Sachdeva, F. R. Cordeiro, V. Belagiannis, I. Reid, and G. Carneiro, "EvidentialMix: Learning with combined open-set and closed-set noisy labels," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3606–3614.

[43] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," 2018, *arXiv:1806.01768*.

[44] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proc. Conf. Adv. Neural Inf. Process. Syst.*, 2016, pp. 1857–1865.

[45] J. Song, Y. Chen, J. Ye, and M. Song, "Spot-adaptive knowledge distillation," *IEEE Trans. Image Process.*, vol. 31, pp. 3359–3370, 2022.

[46] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," 2014, *arXiv:1406.2080*.

[47] R. Xiao et al., "ProMix: Combating label noise via maximizing clean sample utility," 2022, *arXiv:2207.10276*.

[48] W. Wang, J. Shen, and L. Shao, "Video salient object detection via fully convolutional networks," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 38–49, Jan. 2018.

[49] Y. Wang et al., "Iterative learning with open-set noisy labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8688–8696.

[50] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 322–330.

[51] H. Wei, L. Feng, X. Chen, and B. An, "Combating noisy labels by agreement: A joint training method with co-regularization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13726–13735.

[52] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3733–3742.

[53] X. Xia et al., "Are anchor points really indispensable in label-noise learning?" in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019, pp. 6835–6846.

[54] C. Yang, H. Zhou, Z. An, X. Jiang, Y. Xu, and Q. Zhang, "Cross-image relational knowledge distillation for semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 12319–12328.

[55] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4133–4141.

[56] H. Yin et al., "Dreaming to distill: Data-free knowledge transfer via DeepInversion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8715–8724.

[57] X. Yu, B. Han, J. Yao, G. Niu, I. Tsang, and M. Sugiyama, "How does disagreement help generalization against label corruption?" in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 7164–7173.

[58] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," 2016, *arXiv:1612.03928*.

[59] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "*mixup*: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.

[60] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 31, 2018, pp. 8792–8802.

[61] B. Zhao, Q. Cui, R. Song, Y. Qiu, and J. Liang, "Decoupled knowledge distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 11953–11962.

[62] J. Zhu et al., "Complementary relation contrastive distillation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 9260–9269.