Coupled Confusion Correction: Learning from Crowds with Sparse Annotations

Hansong Zhang[1, 2], **Shikun Li**[1, 2], **Dan Zeng**[3], **Chenggang Yan**[4], **Shiming Ge**[1, 2*]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100092, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
[3]Department of Communication Engineering, Shanghai University, Shanghai 200040, China
[4]Hangzhou Dianzi University, Hangzhou 310018, China
{zhanghansong,lishikun,geshiming}@iie.ac.cn, dzeng@shu.edu.cn, cgyan@hdu.edu.cn

Abstract

As the size of the datasets getting larger, accurately annotating such datasets is becoming more impractical due to the expensiveness on both time and economy. Therefore, crowd-sourcing has been widely adopted to alleviate the cost of collecting labels, which also inevitably introduces label noise and eventually degrades the performance of the model. To learn from crowd-sourcing annotations, modeling the expertise of each annotator is a common but challenging paradigm, because the annotations collected by crowd-sourcing are usually highly-sparse. To alleviate this problem, we propose **C**oupled **C**onfusion **C**orrection (CCC), where two models are simultaneously trained to correct the confusion matrices learned by each other. Via bi-level optimization, the confusion matrices learned by one model can be corrected by the distilled data from the other. Moreover, we cluster the "annotator groups" who share similar expertise so that their confusion matrices could be corrected together. In this way, the expertise of the annotators, especially of those who provide seldom labels, could be better captured. Remarkably, we point out that the annotation sparsity not only means the average number of labels is low, but also there are always some annotators who provide very few labels, which is neglected by previous works when constructing synthetic crowd-sourcing annotations. Based on that, we propose to use Beta distribution to control the generation of the crowd-sourcing labels so that the synthetic annotations could be more consistent with the real-world ones. Extensive experiments are conducted on two types of synthetic datasets and three real-world datasets, the results of which demonstrate that CCC significantly outperforms state-of-the-art approaches. Source codes are available at: https://github.com/Hansong-Zhang/CCC.

Introduction

The great success of Deep Neural Networks (DNNs) is much attributed to large-scale and accurately-labeled training datasets. However, collecting such high-quality annotations demands the annotators to be experts in their discipline and is often very time-consuming (Song et al. 2023; Han et al. 2018; Zhang 2022). For instance, the well-known Microsoft COCO dataset were constructed using 20K work hours in total (Lin et al. 2014). To improve the efficiency of construct such datasets, crowd-sourcing (Biel and Gatica-Perez 2013; Buecheler et al. 2010; Han et al. 2019b; Servajean et al. 2017) provides an alternative way to collect annotations in a both time- and economic-efficient manner, where the labels of instances are provided by multiple annotators which could be non-expert. In the paradigm of crowd-sourcing, a large task is usually divided into several mini-tasks (one might overlap another), which will be further distributed to various annotators. On some crowd-sourcing platforms like *Crowdflower* (Wazny 2017) and *Amazon Mechanical Turk*, the annotators can get rewards when a mini-task is accomplished. Although crowd-sourcing is efficient in collecting large-scale annotations, it also inevitably introduces label noise (Sheng, Provost, and Ipeirotis 2008; Snow et al. 2008; Zheng et al. 2017), which is caused by the relatively-low expertise of some of the annotators. On the other hand, the DNNs can eventually memorize the noisy labels due to its great representative capacity (Zhang et al. 2021; Arpit et al. 2017; Ishida et al. 2020). Therefore, increasing the generalization performance of DNNs trained with crowd-sourcing annotations is a challenging problem, which is widely termed as *Learning from Crowds* (LFC).

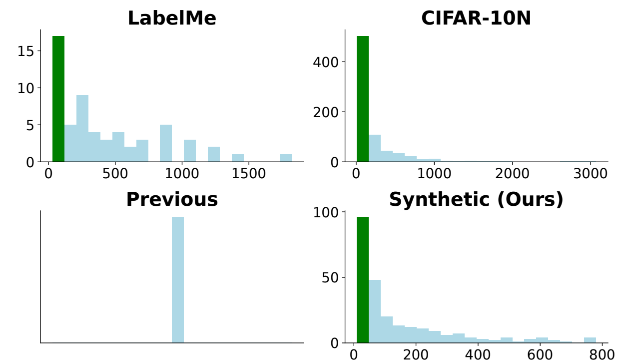


Figure 1: The histogram of the number of labels provided by annotators. The green patch on the left denotes the annotators who only provide seldom labels.

In the discipline of LFC, previous solutions can be roughly divided into two pipelines: **(1)** *Aggregate multiple noisy labels into a more reliable one before or during the training stage.* Among them, the simplest one is Majority

[*]Corresponding Author

Voting (Whitehill et al. 2009; Han et al. 2019a; Schwenker 2013), where all the annotators are assumed to be equally reliable. Although Majority Voting provides a way to alleviate label noises, it is not always feasible to aggregate the labels equally, especially when the task is relatively hard and when most annotators are not reliable enough (Whitehill et al. 2009; Han et al. 2019a; Schwenker 2013). Motivated by that, some advanced methods were designed to model the expertise of each annotator. For instance, Weighted Majority Voting (Li et al. 2014) endows the labels with a weight vector to alleviate the influence of the annotators with low expertise. **(2)** *Training under the supervision of all annotations.* This pipeline of work usually take a set of confusion matrices as a measure of the expertise of annotators (Rodrigues and Pereira 2018; Dawid and Skene 1979; Chu, Ma, and Wang 2021; Wei et al. 2022a). By simultaneously learning the confusion matrices and the classifier in an end-to-end manner, these methods have achieved more promising performance in LFC compared to the label-aggregating ones (Rodrigues and Pereira 2018; Dawid and Skene 1979; Chu, Ma, and Wang 2021; Wei et al. 2022a; Li et al. 2023b).

Challenges. Despite their improved performance, the modeling of the expertise of annotators suffers from the following two problems, which will eventually jeopardises the performance of the classifier. On the one hand, *Annotation Sparsity* is a notorious property of the crowd-sourcing annotations (Ibrahim, Nguyen, and Fu 2023), which means that each annotator only labels a small subset of the whole dataset instead of all of it. On the other hand, the annotator-specific confusion parameters (ASCPs) (e.g. the weight vector or the confusion matrices in the aforementioned two pipelines) of each annotator are only learned based on its individual annotations. Therefore, if one annotator provides too few labels, it is obvious that the supervision of its ASCPs is insufficient and therefore its expertise can not be well-modeled (detailed analysis can be found in the Methodology Section). Note that such scenario is not hypothetical but quite ubiquitous. As shown in Figure 1, in real-world crowd-sourcing datasets LabelMe (Rodrigues and Pereira 2018) and CIFAR-10N (Wei et al. 2022b), a large number of annotators only provide seldom labels, even the authors of CIFAR-10N had enlarged the mini-tasks to alleviate annotation sparsity (Wei et al. 2022b).

To mitigate the sparse annotation problem in LFC, in this paper we propose **C**oupled **C**onfusion **C**orrection (CCC). In our CCC, two simultaneously-trained classifiers are coupled in a way where the confusion matrices learned by one classifier can be corrected by a small **meta set** (a class-balanced set that contains the images and their pseudo labels which are likely to be clean) distilled from the other one. Note that the labels in the meta set are actually from the crowd-sourcing labels. As the meta set has dropped the information of annotator IDs, it can not be directly used to supervise the learning of ASCPs. To this end, we link the meta set and ASCPs via meta learning, in which way each annotator's ASCPs can be corrected by the meta set through bi-level optimization, even if the labels in meta set dose not come from him/her. Moreover, to further alleviate the annotation sparsity, we indicate and justify that there are many annotators who share similar expertise. Based on that founding, we use the K-Means method on the learned confusion matrices to cluster similar annotators so that their confusion matrices can be corrected together.

Our contributions can be summarized as follows:

• **A Meta-Learning-Based Method CCC to Mitigate the Sparse Annotation Problem.** In this work, we show that it is possible to learn one's ASCPs under the supervision of other annotations except for its own individual ones. Specifically, we link the meta set and ASCPs through meta learning paradigm, so that the ASCPs can be corrected by the meta set, enriching its supervision. Moreover, we use K-Means to cluster annotators with similar expertise to further alleviate the annotation sparsity, correcting their ASCPs together.

• **Synthetic More Practical Crowd-Sourcing Annotation using Beta Distribution.** We point out a fact that is neglected by previous works: there exists some few-label annotators (annotators who only provide very few labels) in real-world crowd-sourcing datasets no matter how many the annotators are. To make our synthetic crowd-sourcing annotations more consistent with real-world ones, we propose to use Beta distribution to control the number of labels of each annotator. Previous works naively assumed that the amount of labels are evenly distributed, which may not accurately reflect real-world scenarios. Therefore, our proposed method is useful for future works on LFC when constructing synthetic annotations.

• **Extensive and Comprehensive Experiments.** We conduct experiments on various datasets, which include two types of synthetic crowd-sourcing datasets (independent and dependent cases) and three real-world datasets. The results demonstrate the competitiveness of our proposed method.

Methodology

Notations and Preliminaries. We begin by fixing notations. Suppose that we are given a training dataset $\tilde{\mathcal{D}} = \{\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i\}_{i=1}^N$ which consists of N instances labeled by R annotators in total. Denote the data features as $\mathbf{x} = \{\boldsymbol{x}_i\}_{i=1}^N$ and the crowd-sourcing labels as $\mathbf{y} = \{\tilde{\boldsymbol{y}}_i\}_{i=1}^N$, where $\tilde{\boldsymbol{y}}_i = \left(\tilde{y}_i^1, \tilde{y}_i^2, ..., \tilde{y}_i^R\right)$. The i-th crowd-sourcing label $\tilde{\boldsymbol{y}}_i$ is a R-dimensional vector representing the labels provided by R annotators. Note that it is usually the case that an annotator does not label all the data, so the crowd-sourcing labels may have missing ones. Let C denotes the number of class categories, then the individual label $\tilde{y} \in \{1, 2, ..., C\}$. The goal of *learning from crowds* is to train a classifier $f(\boldsymbol{x}; \boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$ (we take DNN as the classifier in this paper, which takes the instance feature $\boldsymbol{x}$ as input and outputs the possibility distribution over all classes) with training set $\tilde{\mathcal{D}}$, and to make it perform well on the unseen data.

Meta Learning Recap

Next, we will briefly recap the preliminaries on *Meta Learning*. In the meta-learning-based methods (Shu et al. 2019; Wang, Hu, and Hu 2020; Zheng, Awadallah, and Dumais 2021; Xu et al. 2021; Zhang and Pfister 2021), an out-of-sample meta set $\mathcal{D}^{meta} = \{\boldsymbol{x}_j^{meta}, y_j^{meta}\}_{j=1}^M$ is assumed to be available, where y_j^{meta} denotes the true label of the j-th meta data $\boldsymbol{x}_j^{meta}$ and $M (M \ll N)$ denotes the size of meta set. Note that the size of the meta set is much smaller than the training set, hence it is easy to collect in practice. The meta set can be either additionally provided (Shu et al. 2019; Wang, Hu, and Hu 2020; Zheng, Awadallah, and Dumais 2021) or distilled from the training set (Xu et al. 2021; Zhang and Pfister 2021). Various hyper-parameters (e.g. instanced by the transition matrix (Wang, Hu, and Hu 2020), label corrector net (Zheng, Awadallah, and Dumais 2021) or multi-layer perceptron network (Shu et al. 2019)) are called meta parameters and are updated via bi-level optimization, in which the meta set plays an important role in guiding the updating of such meta parameters. Intuitively, the performance of the current model on unseen data is measured by the meta set, the loss of the model on the meta set (termed as meta loss) is minimized in the outer loop of the bi-level optimization so that the gradient can flow back from the meta set to the meta parameters.

Typically, there are two sets of parameters to be optimized in such bi-level optimization, one is the aforementioned meta parameters, the other is the main parameters (mostly instanced by the parameters of the backbone model). Let $\boldsymbol{w}/\boldsymbol{\theta}$ denote the meta / main parameters, and $\mathcal{L}_i^{tra}(\boldsymbol{w}, \boldsymbol{\theta})$ be the loss of the i-th instance in the training set $\tilde{\mathcal{D}}$. The optimal main parameters can be obtained by the training loss:

$$\boldsymbol{\theta}^\star(\boldsymbol{w}) = \arg\min_{\boldsymbol{\theta}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_i^{tra}(\boldsymbol{w}, \boldsymbol{\theta}). \tag{1}$$

Then the meta parameters can be optimized by minimizing the meta loss:

$$\boldsymbol{w}^\star = \arg\min_{\boldsymbol{w}} \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}_j^{meta}(\boldsymbol{\theta}^\star(\boldsymbol{w})), \tag{2}$$

where the $\mathcal{L}_j^{meta}(\boldsymbol{\theta}^\star(\boldsymbol{w}))$ is the loss of the j-th instance in $\mathcal{D}^{meta}$. However, directly solving Eq. (1) and Eq. (2) is impractical, because the optimal $\boldsymbol{\theta}$ needs to be calculated whenever the meta parameters $\boldsymbol{w}$ is updated. Therefore, an online optimization method, as a trade-off between the feasibility and effectiveness, has been widely applied to the optimization of meta learning, which consists of three stages, including *Virtual*, *Meta*, and *Actual* stage (Wang, Hu, and Hu 2020). The optimization is performed in a mini-batch gradient descent manner, in t-th iteration, a training mini-batch $\tilde{\mathcal{M}} = \{\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i\}_{i=1}^n$ and a meta mini-batch $\mathcal{M}^{meta} = \{\boldsymbol{x}_j^{meta}, y_j^{meta}\}_{j=1}^m$ is fetched, where the n and m is the batch size of training and meta set respectively. Let η_v, η_m, η_a denote the learning rates in the *Virtual, Meta,* and *Actual* stages respectively. In *Virtual* Stage, the main parameters is "virtually" updated by:

$$\hat{\boldsymbol{\theta}}_t = \boldsymbol{\theta}_{t-1} - \eta_v \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}_{t-1}} \mathcal{L}_i^{tra}(\boldsymbol{w}_{t-1}, \boldsymbol{\theta}_{t-1}). \tag{3}$$

This stage is named "virtual" because the true main parameters are kept unchanged, the above $\hat{\boldsymbol{\theta}}_t$ is fixed and only used to update the meta parameters in the following *Meta* stage:

$$\boldsymbol{w}_t = \boldsymbol{w}_{t-1} - \eta_m \frac{1}{m} \sum_{j=1}^{m} \nabla_{\boldsymbol{w}_{t-1}} \mathcal{L}_j^{meta}(\boldsymbol{w}_{t-1}, \hat{\boldsymbol{\theta}}_t). \tag{4}$$

The *Virtual* stage, together with the *Meta* stage, can be deem as the outer loop of the bi-level optimization, which updates the meta parameters under the supervision of both meta set and training set (Wang, Hu, and Hu 2020). The "virtual" main parameters $\hat{\boldsymbol{\theta}}_t$ is deleted as soon as the *Meta* stage finished. Last, the $\boldsymbol{w}_t$ that comes from the outer loop will be fixed to update the main parameters in the *Actual* stage:

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta_a \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}_{t-1}} \mathcal{L}_i^{tra}(\boldsymbol{w}_t, \boldsymbol{\theta}_{t-1}). \tag{5}$$

Unlike *Virtual* stage, the main parameters in this stage is "actually" updated. After the above three stages, both the main and meta parameters are updated and ready for the $(t+1)$-th iteration. Note that the *Actual* stage serves as the inner loop of the bi-level optimization. By alternating the outer and inner loop, the optimal main and meta parameters can be derived.

How Annotation Sparsity Affects Training

In this section, we would like to show why the sparse annotation phenomenon is harmful, which motivates our work. For notation convenience, we use a matrix $\mathcal{A}_{(N\times R)}$ to indicate the annotation presence, i.e., $\mathcal{A}_{i,j}$ is True if the i-th instance is labeled by j-th annotator, and False otherwise. Without loss of generality, we hereby take CrowdLayer (Rodrigues and Pereira 2018) as an example, which uses R confusion matrices $\{\boldsymbol{T}^r\}_{r=1}^R$ to model the annotator-specific expertise[1]. In CrowdLayer, the parameters of backbone DNN $\boldsymbol{\theta}$ and confusion matirces $\{\boldsymbol{T}^r\}_{r=1}^R$ are obtained by computing the following loss:

$$\boldsymbol{\theta}^\star, \{\boldsymbol{T}^{r\star}\}_{r=1}^R = \arg\min_{\boldsymbol{\theta}, \{\boldsymbol{T}^r\}_{r=1}^R} \frac{1}{N} \sum_{i=1}^{N} \sum_{r=1}^{R} \mathbb{1}[\mathcal{A}_{i,r}] \mathcal{L}_{i,r}(\boldsymbol{T}^r, \boldsymbol{\theta}), \tag{6}$$

where $\mathbb{1}[\cdot]$ denotes the indicator function, which outputs 1 if the event is True, and 0 otherwise. $\mathcal{L}_{i,r}(\boldsymbol{T}^r, \boldsymbol{\theta}) = \ell(\boldsymbol{T}^r f(\boldsymbol{x}_i|\boldsymbol{\theta}), \tilde{y}_i^r)$ stands for the loss of i-th instance and r-th annotator in CrowdLayer, where ℓ is a loss function such as widely-used cross entropy and f is the softmax output of the classifier. For the r_0-th confusion matrix $\boldsymbol{T}^{r_0}$ (i.e. the ASCPs of r_0-th annotator), by modifying Eq. (6), the gradient w.r.t. $\boldsymbol{T}^{r_0}$ (denoted by $\boldsymbol{g}^{r_0}$) can be derived as:

$$\begin{aligned} \boldsymbol{g}^{r_0} &= \nabla_{\boldsymbol{T}^{r_0}} \frac{1}{N} \sum_{i=1}^{N} \sum_{r=1}^{R} \mathbb{1}[\mathcal{A}_{i,r}] \mathcal{L}_{i,r}(\boldsymbol{T}^r, \boldsymbol{\theta}) \\ &= \nabla_{\boldsymbol{T}^{r_0}} \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[\mathcal{A}_{i,r_0}] \mathcal{L}_{i,r_0}(\boldsymbol{T}^{r_0}, \boldsymbol{\theta}), \end{aligned} \tag{7}$$

from which we notice that only the instances labeled by r_0-th annotator provide supervised information to its own confusion matrix, let alone the labels could be noisy. Therefore,

[1]The analyses also applies to the methods with other ASCPs.

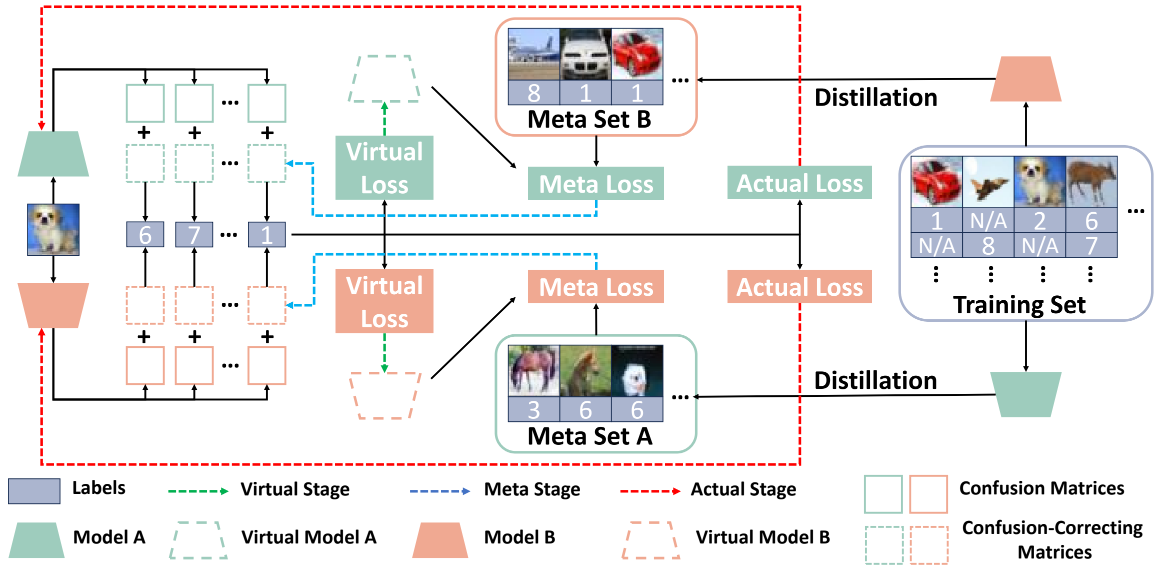
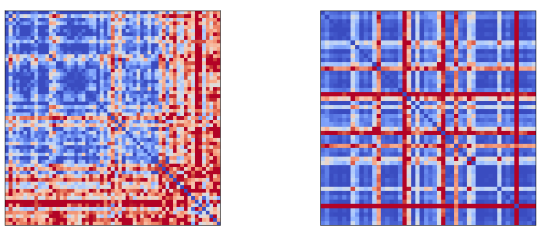


Figure 2: The illustration of the our method. "N/A" means the label is missing.



(a) LabelMe #1∼59 (b) CIFAR-10N #50∼100

Figure 3: The distances of true confusion matrices of every two annotators in LabelMe and CIFAR-10N datasets. The # denotes the index of annotators. The more blue (red) the pixel is, the smaller (larger) value it is.

if r_0-th annotator labels too few data, its confusion matrix would be poorly learned. It is worth noting that Figure 1 can be deemed as the histogram of $\sum_{i=1}^{N} \mathbb{1}[\mathcal{A}_{i,r}]$, which highlights that a significant proportion of annotators contribute very few labels. This observation suggests that the issue mentioned above is prevalent in real-world scenarios.

Coupled Confusion Correction

Before diving into technical details, we first claim and justify an assumption that: *In real-world crowd-sourcing datasets, there are various annotators who share similar expertise.*

We hereby take two common real-world crowd-sourcing datasets as examples, including LabelMe and CIFAR-10N (Rodrigues and Pereira 2018; Wei et al. 2022b; Chu, Ma, and Wang 2021). We calculate the true confusion matrices of each annotators $(CM^r)_{(C\times C)}$ in such two datasets as follows:

$$CM^r_{p,q} = \begin{cases} 0, & \text{if} \sum_{i=1}^{N} \mathbb{1}[\mathcal{A}_{i,r} \cap y_i = p] = 0 \\ \dfrac{\sum_{i=1}^{N} \mathbb{1}[\mathcal{A}_{i,r} \cap y_i = p \cap \tilde{y}_i^r = q]}{\sum_{i=1}^{N} \mathbb{1}[\mathcal{A}_{i,r} \cap y_i = p]}, & \text{else} \end{cases}$$

where y_i denotes the true label of i-th instance. After getting the true confusion matrices, we calculate the distance between the true confusion matrices of every two annotators using mean-square-error. Figure 3 illustrates the confusion distances between annotators in the LabelMe and CIFAR-10N datasets, where blue and red indicate small and large distances, respectively. As we can tell, there are many "blue patches" in the distance matrices, which suggests that the confusion matrices of annotators in such patch is similar to each other. Therefore, if their expertise can be considered together, the annotation sparsity can also be alleviated to some extent.

Overview of Our Approach. In this paper, to alleviate the annotation sparsity, we propose **C**oupled **C**onfusion **C**orrection (CCC), where we simultaneously train two models and utilize the meta set distilled to correct the ASCPs learned by each other. Following (Cao et al. 2019; Chu, Ma, and Wang 2021; Rodrigues and Pereira 2018; Albarqouni et al. 2016), we also adopt the confusion matrices to model the annotators' expertise. A set of confusion-correcting-matrices $\{\boldsymbol{V}^r\}_{r=1}^{G}$ are regarded as meta parameters to correct the confusion matrices of annotators $\{\boldsymbol{T}^r\}_{r=1}^{R}$, where G is a hyper parameter indicating how many similar groups of annotators are. We run the K-Means (Forgy 1965; Arthur and Vassilvitskii 2007) method on the learned confusion matrices $\{\boldsymbol{T}^r\}_{r=1}^{R}$ to obtain their group indexes at each epoch. Unlike CrowdLayer (Rodrigues and Pereira 2018), which only updates the confusion matrices based on individual annotations, CCC allows the confusion matrix of one annotator to be updated under the supervision of both his/her group's annotations and the meta set. This makes CCC more resilient to annotation sparsity, leading to improved classifier performance. In the following, we first explain why the supervised information can flow back from the meta set to the confusion-correcting-matrices. Then we present the distillation strategy of the meta set and the training objectives of different stages in our CCC. The framework of our method is illustrated in Figure 2 and the pseudo code can be found in the Appendix.

Correcting Confusion Matrices by Meta Set. In our CCC, the corrected confusion matrices are obtained through the following bi-level optimization:

$$\{\boldsymbol{V}^{r\star}\}_{r=1}^{G} = \mathop{\arg\min}_{\{\boldsymbol{V}^r\}_{r=1}^{G}} \frac{1}{M}\sum_{j=1}^{M}\mathcal{L}_j^{meta}(\boldsymbol{\theta}^\star), \text{s.t.},$$
$$\boldsymbol{\theta}^\star, \{\boldsymbol{T}^{r\star}\}_{r=1}^{R} = \mathop{\arg\min}_{\boldsymbol{\theta}, \{\boldsymbol{T}^r\}_{r=1}^{R}} \frac{1}{N}\sum_{i=1}^{N}\sum_{r=1}^{R}\mathbb{1}[\mathcal{A}_{i,r}]\mathcal{L}_{i,r}^{tra}(\boldsymbol{T}_{cor}^{r}, \boldsymbol{\theta}) \tag{8}$$

where $\boldsymbol{T}_{cor}^{r} = \boldsymbol{T}^r + \boldsymbol{V}^{g(r)}$ denotes the corrected confusion matrix of r-th annotator with $g(r)$ representing the index of group which the r-th annotator belongs to. Let ℓ denotes the cross entropy loss function, the meta loss here is then defined as $\mathcal{L}_j^{meta}(\boldsymbol{\theta}^\star) = \ell(f(\boldsymbol{x}_j^{meta}|\boldsymbol{\theta}^\star), y_j^{meta})$, and the training loss is $\mathcal{L}_{i,r}^{tra}(\boldsymbol{T}_{cor}^{r}, \boldsymbol{\theta}) = \ell((\boldsymbol{T}^r + \boldsymbol{V}^{g(r)})f(\boldsymbol{x}_i|\boldsymbol{\theta}), \tilde{y}_i^r)$. Next, let us focus on the gradient w.r.t. the confusion-correcting-matrix of r_0-th annotator $\boldsymbol{V}^{g(r_0)}$ (denoted by $\boldsymbol{g}_{cor}^{r_0}$):

$$\begin{aligned}
\boldsymbol{g}_{cor}^{r_0} &= \nabla_{\boldsymbol{V}^{g(r_0)}} \frac{1}{M}\sum_{j=1}^{M}\mathcal{L}_j^{meta}(\boldsymbol{\theta}^\star) = \frac{1}{M}\sum_{j=1}^{M}\frac{\partial \mathcal{L}_j^{meta}}{\partial \boldsymbol{\theta}^\star}\cdot\frac{\partial \boldsymbol{\theta}^\star}{\partial \boldsymbol{V}^{g(r_0)}} \\
&= \frac{1}{M}\sum_{j=1}^{M}\frac{\partial \mathcal{L}_j^{meta}}{\partial \boldsymbol{\theta}^\star}\cdot\frac{1}{N}\sum_{i=1}^{N}\sum_{r=1}^{R}\mathbb{1}[\mathcal{A}_{i,r}]\nabla^2_{\boldsymbol{\theta},\boldsymbol{V}^{g(r_0)}}\mathcal{L}_{i,r}^{tra} \\
&= \frac{1}{MN}\sum_{j=1}^{M}\frac{\partial \mathcal{L}_j^{meta}}{\partial \boldsymbol{\theta}^\star}\cdot\sum_{i=1}^{N}\mathbb{1}[\mathcal{A}_{i,r_0}]\nabla^2_{\boldsymbol{\theta},\boldsymbol{V}^{g(r_0)}}\mathcal{L}_{i,r_0}^{tra}.
\end{aligned} \tag{9}$$

Comparing Eq. (7) and Eq. (9), we can see that every data in the meta set contribute to the overall gradient of the confusion-correcting-matrix of the r_0-th annotator. In this way, the learned confusion matrices can be corrected under the supervision of the meta set. Remarkably, because we cluster annotators with similar expertise, the confusion-correcting-matrices of two annotators (r_0, r_1) can be updated together as long as they belong to the same group $(g(r_0) = g(r_1))$, which further hinders the influence of annotation sparsity.

Distilling Meta Sets using Two Models. In most previous meta learning approaches (Shu et al. 2019; Wang, Hu, and Hu 2020), the meta set is demanded to be provided in addition to the training set, which may harms the fairness of the comparison to baselines. To this end, FaMUS (Xu et al. 2021) and FSR (Zhang and Pfister 2021) provided another way to construct meta set, where it can be distilled directly from the training set. However, their methods are constrained in the single-label scenario, hence we propose a new distilling strategy to match the crowd-sourcing annotations. The algorithm is quite simple to understand, as many sample-selection-based methods pointed out (Arpit et al. 2017; Han et al. 2018; Li, Socher, and Hoi 2020), the samples with smaller loss is more likely to be clean. Therefore, for each class c, we first fetch all the instances that are attached with the label c, then we select the ones with M/C smallest losses. Nevertheless, it is worth noting that in Eq. 8, if the losses of the instances that we select are already small, the supervised information that we obtain from the meta set may also be insufficient. Consequently, we simultaneously train two models to distill meta set for each other, in which way we can mitigate the confirmation bias (i.e., one model would accumulate its errors (Li, Socher, and Hoi 2020)) to ensure the meta set to be sufficiently informative while maintaining clean.

Training Objective. As stated above, in iteration t, we have a training mini-batch $\tilde{\mathcal{M}} = \{\boldsymbol{x}_i, \tilde{\boldsymbol{y}}_i\}_{i=1}^{n}$ and a meta mini-batch $\mathcal{M}^{meta} = \{\boldsymbol{x}_j^{meta}, y_j^{meta}\}_{j=1}^{m}$ for each model. The training objective and outcomes in three stages (*virtual, meta, actual*) are shown as follows:

$$\text{Virtual}: \frac{1}{n}\sum_{i=1}^{n}\sum_{r=1}^{R}\mathcal{L}_{i,r}^{tra}((\boldsymbol{T}_{t-1}^{r} + \boldsymbol{V}_{t-1}^{g(r)}), \boldsymbol{\theta}_{t-1}) \rightarrow \hat{\boldsymbol{\theta}}_t \tag{10}$$

$$\text{Meta}: \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}_j^{meta}(\hat{\boldsymbol{\theta}}_t) \rightarrow \{\boldsymbol{V}_t^r\}_{r=1}^{G} \tag{11}$$

Note that the confusion-correcting-matrices are initialized as zero matrices at the beginning of each iteration, after the above two stages (i.e. the outer loop), we have obtained the corrected confusion matrices $\boldsymbol{T}_{cor}^{r}|_{t-1} = \boldsymbol{T}_{t-1}^{r} + \boldsymbol{V}_t^{g(r)}$, which can be used in the following *actual* stage (i.e. the inner loop):

$$\text{Actual}: \frac{1}{n}\sum_{i=1}^{n}\sum_{r=1}^{R}\mathcal{L}_{i,r}^{tra}(\boldsymbol{T}_{cor}^{r}|_{t-1}, \boldsymbol{\theta}_{t-1}) \rightarrow \{\boldsymbol{T}_t^r\}_{r=1}^{R}, \boldsymbol{\theta}_t \tag{12}$$

Experiments

We conduct experiments on two types of synthetic datasets (independent and correlated confusion) and three real-world datasets. To make the results reliable, all experiments are repeated three times with different random seeds, both the best accuracy on test set during training and the accuracy at the last epoch is reported. All experiments are conducted on a single NVIDIA RTX 3090 GPU. Due to space limitation, we only present the brief descriptions of datasets and results in the main paper, more information about datasets and the implementation details can be found in the Appendix.

Baselines. We compare our method with extensive state-of-the-art approaches. The compared approaches include: (1) Majority Vote (Whitehill et al. 2009). (2) Crowd Layer (Rodrigues and Pereira 2018). (3) Doctor Net (Guan et al. 2018).

Dataset		CIFAR-10				Fashion-MNIST			
Method / Case		IND-I	IND-II	IND-III	IND-IV	IND-I	IND-II	IND-III	IND-IV
MajorVote	Best	55.08±0.38	47.85±2.20	54.27±1.91	66.94±1.01	84.89±2.24	79.94±2.30	76.72±1.00	89.03±0.75
	Last	41.14±0.37	32.90±0.71	42.00±0.39	60.18±1.95	61.00±0.65	51.31±0.56	56.88±0.29	78.02±0.52
CrowdLayer	Best	63.06±1.49	53.50±1.24	59.39±1.21	69.09±0.62	88.83±0.61	84.77±2.09	87.01±3.05	90.31±0.42
	Last	54.78±0.49	43.71±0.62	53.17±0.34	67.51±0.91	75.89±0.19	66.39±0.21	66.67±0.47	84.67±0.20
DoctorNet	Best	69.63±0.86	63.43±1.46	69.77±2.02	76.25±0.47	90.54±0.23	88.78±0.68	88.29±1.20	90.96±0.32
	Last	66.80±1.12	57.95±1.09	65.40±0.98	74.60±0.45	83.01±0.13	76.07±0.49	75.77±0.20	86.78±0.19
Max-MIG	Best	71.63±1.15	65.52±0.99	69.73±1.09	75.97±0.39	91.03±0.38	90.15±0.45	90.07±0.49	91.32±0.25
	Last	66.47±0.48	56.63±0.46	61.68±0.63	73.19±0.59	82.16±0.32	73.33±0.39	73.53±0.47	86.16±0.37
CoNAL	Best	67.28±1.24	59.27±1.53	65.93±0.15	78.16±0.27	89.95±0.30	87.62±0.88	89.13±0.44	91.72±0.14
	Last	56.18±0.30	47.02±0.65	57.16±0.49	71.96±0.53	75.27±0.17	65.80±0.32	71.16±0.29	86.67±0.25
UnionNet	Best	74.67±0.82	68.92±0.96	71.52±1.14	79.29±0.65	91.30±0.37	90.09±0.42	89.67±0.58	91.61±0.25
	Last	74.44±0.60	63.41±0.75	67.00±0.42	78.78±0.74	88.22±0.19	78.29±0.29	79.09±0.54	90.56±0.21
CCC (Ours)	Best	**80.16±0.23**	**75.33±0.43**	**78.28±0.25**	**83.06±0.26**	**92.59±0.01**	**91.93±0.18**	**92.50±0.13**	**93.23±0.06**
	M.I.	↑ 5.49	↑ 6.41	↑ 6.76	↑ 3.77	↑ 1.29	↑ 1.78	↑ 2.43	↑ 1.51
	Last	**79.50±0.12**	**73.52±0.39**	**77.20±0.26**	**82.56±0.07**	**92.51±0.06**	**91.58±0.39**	**92.17±0.04**	**93.14±0.06**
	M.I.	↑ 5.06	↑ 10.11	↑ 10.20	↑ 3.78	↑ 4.29	↑ 13.29	↑ 13.08	↑ 2.58
Method / Case		COR-I	COR-II	COR-III	COR-IV	COR-I	COR-II	COR-III	COR-IV
MajorVote	Best	59.18±3.38	63.26±1.39	67.03±1.00	67.39±1.63	85.73±1.51	85.22±1.99	89.38±0.38	89.25±0.70
	Last	43.57±0.61	50.36±0.76	59.04±0.62	57.97±1.07	60.83±0.57	66.11±0.83	79.69±0.77	75.68±0.46
CrowdLayer	Best	62.51±1.12	67.45±1.03	69.24±0.69	70.07±0.40	88.49±0.44	89.87±0.17	90.19±0.28	89.88±0.23
	Last	54.61±0.52	62.70±0.41	66.27±0.28	68.64±0.50	76.02±0.20	79.15±0.17	85.33±0.09	83.93±0.19
DoctorNet	Best	67.30±1.28	74.01±0.62	73.87±0.48	77.52±0.49	89.29±0.37	90.54±0.32	90.36±0.21	90.86±0.21
	Last	63.02±0.92	71.48±1.22	72.19±0.41	76.80±0.59	82.38±0.17	84.78±0.37	86.65±0.18	87.28±0.16
Max-MIG	Best	71.69±0.68	74.55±0.56	76.83±0.30	75.81±0.68	90.27±0.07	91.26±0.17	91.38±0.16	91.17±0.28
	Last	67.19±0.33	73.28±0.60	75.82±0.46	73.94±0.60	81.20±0.33	84.58±0.49	88.14±0.21	84.83±0.49
CoNAL	Best	69.33±0.46	72.21±0.30	75.97±0.19	79.82±0.49	89.48±0.34	90.79±0.40	91.37±0.10	92.02±0.14
	Last	58.24±0.63	64.10±0.27	68.42±0.65	73.82±0.83	75.49±0.30	78.36±0.28	85.93±0.40	88.47±0.47
UnionNet	Best	74.46±0.43	78.13±0.43	80.02±0.51	80.99±0.66	90.83±0.40	91.52±0.14	91.94±0.22	92.31±0.22
	Last	72.84±0.31	77.58±0.47	79.37±0.62	80.85±0.66	85.32±0.21	88.60±0.37	91.82±0.41	92.24±0.32
CCC (Ours)	Best	**80.77±0.10**	**81.84±0.14**	**83.32±0.13**	**82.72±0.11**	**91.93±0.09**	**92.87±0.06**	**93.01±0.07**	**93.18±0.10**
	M.I.	↑ 6.31	↑ 3.71	↑ 3.30	↑ 1.73	↑ 1.10	↑ 1.35	↑ 1.07	↑ 0.87
	Last	**80.07±0.22**	**81.50±0.09**	**83.07±0.13**	**82.53±0.20**	**91.75±0.11**	**92.70±0.07**	**92.88±0.07**	**93.10±0.09**
	M.I.	↑ 7.23	↑ 3.92	↑ 3.70	↑ 1.68	↑ 6.43	↑ 4.10	↑ 1.06	↑ 0.86

Table 1: Comparison with state-of-the-art approaches in test accuracy (%) on synthetic datasets with independent (IND) and correlated (COR) confusions. The best results are in bold. The "M.I." represents minimal improvement.

(4) Max-MIG (Cao et al. 2019). (5) CoNAL (Chu, Ma, and Wang 2021). (6) UnionNet (Wei et al. 2022a).

Evaluation on Synthetic Datasets

Datasets. We evaluate our method on a variety of synthetic datasets, which are derived from two widely used datasets, i.e., CIFAR-10 (Krizhevsky, Hinton et al. 2009) and Fashion-MNIST (Xiao, Rasul, and Vollgraf 2017). In general, we divide the synthetic datasets into two types: 1) *independent confusion*, where the confusions of annotators are independent from each other and 2) *correlated confusion*, where the confusions among different annotators may affect each other. Due to the limited page, we provide the generation details of these two confusions in the Appendix.

Results. Table 1 shows the results on synthetic datasets with independent and correlated confusions. As shown, our CCC achieves superior performance than state-of-the-art methods across all scenarios. Based on the minimal improvement (M.I.) results, it is evident that the M.I. achieved at the last epoch is notable, which suggests that our CCC effectively prevents over-fitting while improving the model's best performance. On the other hand, the common baseline approach Majority Voting performs worst consistently, and the accuracy at last epoch decreases a lot compared to the best one, which is consistent with other works (Cao et al. 2019; Chu, Ma, and Wang 2021; Li et al. 2023a; Rodrigues and Pereira 2018; Li et al. 2020). This is because the naive majority vote fails to consider the difference among confusions of annotators, making the DNN eventually over-fits to the noisy labels. As for the approaches that consider annotators differently, their results are unstable, none of them performs better than any other methods consistently. The accuracy drop is also observed between the best and last epochs, the reason of which lies within that the ASCPs (e.g. confusion matrices in (Rodrigues and Pereira 2018; Cao et al. 2019)) of some annotators are poorly modeled, which makes the model finally memorize the noisy labels. It is worth noting that without using meta set to correct the learned confusion matrices, CCC will reduce to CrowdLayer (Rodrigues and Pereira 2018). With the use of the meta set, the results of our CCC is significantly improved compared to that of CrowdLayer (Rodrigues and Pereira 2018). This suggests that the confusion

Method / Dataset		LabelMe	CIFAR-10N	MUSIC
MajorVote	Best	80.72±0.45	81.08±0.34	67.67±0.98
	Last	77.19±1.18	80.89±0.25	62.00±1.05
CrowdLayer	Best	84.01±0.36	80.43±0.25	71.67±0.50
	Last	82.32±0.41	80.14±0.28	69.33±0.93
DoctorNet	Best	82.32±0.49	83.68±0.33	67.33±0.74
	Last	81.73±0.49	83.32±0.29	65.33±0.75
Max-MIG	Best	86.28±0.35	83.25±0.47	75.33±0.69
	Last	83.16±0.66	83.12±0.63	71.67±0.93
CoNAL	Best	87.46±0.53	82.51±0.15	74.00±0.89
	Last	**84.85±0.91**	80.92±0.28	68.67±1.88
UnionNet	Best	85.19±0.42	82.66±1.12	72.33±0.58
	Last	82.66±0.38	82.32±1.09	68.33±0.91
CCC (Ours)	Best	**87.65±1.10**	**86.36±0.05**	**76.22±0.42**
	M.I.	↑ 0.19	↑ 2.68	↑ 0.89
	Last	84.79±0.80	**86.12±0.12**	**72.89±0.68**
	M.I.	↓ 0.06	↑ 2.80	↑ 1.22

Table 2: Comparison with state-of-the-art approaches in test accuracy (%) on real-world datasets. The best results are in bold. The "M.I." represents minimal improvement.

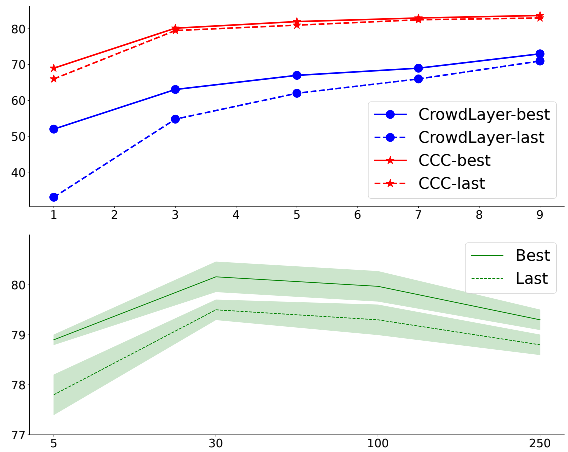


Figure 4: The ablation studies on sparsity level (top) and number of annotator groups (bottom). The horizontal axis represents average number of annotators (top) and number of annotator groups (bottom), while the vertical axis denotes test accuracy.

matrices employed in CCC are better able to model the expertise of annotators.

Evaluation on Real-World Datasets

Datasets. Three real-world datasets are adopted for evaluation, including LabelMe (Rodrigues and Pereira 2018), CIFAR-10N (Wei et al. 2022b) and MUSIC (Rodrigues, Pereira, and Ribeiro 2014). The first two datasets are image classification datasets, while the last one is a music genre classification dataset. The annotations of these real-world datasets are collected on *Amazon Mechanical Turk*.
Results. The evaluation results on three real-world datasets are presented in Table 2. As shown, our CCC outperforms other methods in two metrics (best and last) on CIFAR-10N and MUSIC. Additionally, on the LabelMe dataset, our CCC yields a new state-of-the-art result in terms of the best accuracy during training and achieves competitive accuracy results at the last epoch. These results provide evidence that our CCC is effective in handling real-world crowd-sourced annotations. Remarkably, the results of our CCC on CIFAR-10N are even comparable to the results that were trained with clean labels (refer to the Appendix). This means that the expertise of annotators in the CIFAR-10N dataset has been fully captured by our CCC.

Ablation Study

Annotation Sparsity. To study the performance of CCC under different sparsity levels, we conduct experiments under different average numbers of annotations. To this end, we synthesize 4 more sets of annotations for the CIFAR-10 (Krizhevsky, Hinton et al. 2009) benchmark with an average number of annotations of 1,5,7,9, the confusion patterns of these datasets are identical to CIFAR-10-IND-I. The first chart in Figure 4 depicts the corresponding results compared to CrowdLayer (Rodrigues and Pereira 2018), from which we can see that our CCC yields higher test accuracy in both best and last metrics. Remarkably, even in the sparsest case, our CCC can still bring significant improvement (17.15% on best and 32.95% on last).

Number of Annotator Groups. As shown in Figure 3, there are various annotator groups in crowd-sourcing datasets, hence it is not necessary to consider all the annotators separately. However, the number of annotator groups is hard to identify, because the true labels are not available in practice. Therefore, in our CCC, the number of annotator groups G is deemed as a hyper-parameter. We conduct the ablation study of G on the CIFAR-10-IND-I dataset. The second chart in Figure 4 demonstrates the test accuracy (%) over different G's. If G is small, distinct annotators may be grouped together, leading to potential interference between their annotations. Conversely, if G is large, similar annotators may be considered separately, resulting in poor modeling of their expertise due to annotation sparsity.

Conclusions

In this work, we proposed a novel framework called **C**oupled **C**onfusion **C**orrection (CCC) to learn from multiple annotators, where the confusion matrices learned by one model can be corrected by the meta set distilled from the other. To further alleviate annotation sparsity, we unite various annotators based on their expertise using K-Means so that their confusion matrices could be corrected together. Besides, we reveal a common but neglected phenomenon in crowd-sourcing datasets that there are always annotators who provide seldom labels, which hinders the modeling of their expertise. Through extensive experiments across both synthetic and real-world datasets, we show that our CCC significantly outperforms other state-of-the-art approaches.

Acknowledgments

This work was partially supported by grants from the National Key Research and Development Plan (2020AAA0140001), Beijing Natural Science Foundation (19L2040), and Open Research Project of National Key Laboratory of Science and Technology on Space-Born Intelligent Information Processing (TJ-02-22-01).

References

Albarqouni, S.; Baur, C.; Achilles, F.; Belagiannis, V.; Demirci, S.; and Navab, N. 2016. AggNet: Deep Learning From Crowds for Mitosis Detection in Breast Cancer Histology Images. *IEEE TMI*, 35(5): 1313–1321.

Arpit, D.; Jastrzebski, S.; Ballas, N.; et al. 2017. A Closer Look at Memorization in Deep Networks. In *ICML*, 233–242.

Arthur, D.; and Vassilvitskii, S. 2007. K-means++ the advantages of careful seeding. In *ACM-SIAM SODA*, 1027–1035.

Biel, J.; and Gatica-Perez, D. 2013. The YouTube Lens: Crowdsourced Personality Impressions and Audiovisual Analysis of Vlogs. *IEEE TMM*, 15(1): 41–55.

Buecheler, T.; Sieg, J. H.; Füchslin, R. M.; and Pfeifer, R. 2010. Crowdsourcing, Open Innovation and Collective Intelligence in the Scientific Method - A Research Agenda and Operational Framework. In *ALIFE*, 679–686.

Cao, P.; Xu, Y.; Kong, Y.; and Wang, Y. 2019. Max-MIG: an Information Theoretic Approach for Joint Learning from Crowds. In *ICLR*.

Chu, Z.; Ma, J.; and Wang, H. 2021. Learning from Crowds by Modeling Common Confusions. In *AAAI*, 5832–5840.

Dalvi, N. N.; Dasgupta, A.; Kumar, R.; and Rastogi, V. 2013. Aggregating crowdsourced binary ratings. In *WWW*, 285–294.

Dawid, A. P.; and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *JRSS-C*, 28(1): 20–28.

Forgy, E. W. 1965. Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications. *Biometrics*, 21: 768–769.

Guan, M. Y.; Gulshan, V.; Dai, A. M.; and Hinton, G. E. 2018. Who Said What: Modeling Individual Labelers Improves Classification. In *AAAI*, 3109–3118.

Han, B.; Tsang, I. W.; Chen, L.; Zhou, J. T.; and Yu, C. P. 2019a. Beyond Majority Voting: A Coarse-to-Fine Label Filtration for Heavily Noisy Labels. *IEEE TNNLS*, 30(12): 3774–3787.

Han, B.; Yao, Q.; Pan, Y.; Tsang, I. W.; Xiao, X.; Yang, Q.; and Sugiyama, M. 2019b. Millionaire: a hint-guided approach for crowdsourcing. *MLJ*, 108(5): 831–858.

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I. W.; and Sugiyama, M. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 8536–8546.

Ibrahim, S.; Nguyen, T.; and Fu, X. 2023. Deep Learning From Crowdsourced Labels: Coupled Cross-Entropy Minimization, Identifiability, and Regularization. In *ICLR*.

Ipeirotis, P. G.; Provost, F. J.; and Wang, J. 2010. Quality management on Amazon Mechanical Turk. In *ACM SIGKDD*, 64–67.

Ishida, T.; Yamane, I.; Sakai, T.; Niu, G.; and Sugiyama, M. 2020. Do We Need Zero Training Loss After Achieving Zero Training Error? In *ICML*, 4604–4614.

Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparameterization with Gumbel-Softmax. In *ICLR*.

Jones, D. R. 2001. A Taxonomy of Global Optimization Methods Based on Response Surfaces. *J. Glob. Optim.*, 21(4): 345–383.

Karger, D. R.; Oh, S.; and Shah, D. 2011. Budget-optimal crowdsourcing using low-rank matrix approximations. In *ACCCC*, 284–291.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, J.; Socher, R.; and Hoi, S. C. H. 2020. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In *ICLR*.

Li, Q.; Li, Y.; Gao, J.; Zhao, B.; Fan, W.; and Han, J. 2014. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *ACM SIGMOD*, 1187–1198.

Li, S.; Ge, S.; Hua, Y.; Zhang, C.; Wen, H.; Liu, T.; and Wang, W. 2020. Coupled-View Deep Classifier Learning from Multiple Noisy Annotators. In *AAAI*, 4667–4674.

Li, S.; Liu, T.; Tan, J.; Zeng, D.; and Ge, S. 2023a. Trustable Co-Label Learning From Multiple Noisy Annotators. *IEEE TMM*, 25: 1045–1057.

Li, S.; Xia, X.; Deng, J.; Ge, S.; and Liu, T. 2023b. Transferring Annotator-and Instance-dependent Transition Matrix for Learning from Crowds. *arXiv*.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*, 740–755.

Maclaurin, D.; Duvenaud, D.; and Adams, R. P. 2015. Gradient-based Hyperparameter Optimization through Reversible Learning. In *ICML*, 2113–2122.

Paszke, A.; Gross, S.; Massa, F.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 8024–8035.

Peterson, J. C.; Battleday, R. M.; Griffiths, T. L.; and Russakovsky, O. 2019. Human Uncertainty Makes Classification More Robust. In *ICCV*, 9616–9625.

Rodrigues, F.; and Pereira, F. C. 2018. Deep Learning from Crowds. In *AAAI*, 1611–1618.

Rodrigues, F.; Pereira, F. C.; and Ribeiro, B. 2014. Gaussian Process Classification and Active Learning with Multiple Annotators. In *ICML*, 433–441.

Schwenker, F. 2013. Ensemble Methods: Foundations and Algorithms [Book Review]. *IEEE CIM*, 8(1): 77–79.

Servajean, M.; Joly, A.; Shasha, D. E.; Champ, J.; and Pacitti, E. 2017. Crowdsourcing Thousands of Specialized Labels: A Bayesian Active Training Approach. *IEEE TMM*, 19(6): 1376–1391.

Sheng, V. S.; Provost, F. J.; and Ipeirotis, P. G. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *ACM SIGKDD*, 614–622.

Shu, J.; Xie, Q.; Yi, L.; Zhao, Q.; Zhou, S.; Xu, Z.; and Meng, D. 2019. Meta-Weight-Net: Learning an Explicit Mapping For Sample Weighting. In *NeurIPS*, 1917–1928.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*.

Snow, R.; O'connor, B.; Jurafsky, D.; and Ng, A. Y. 2008. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *EMNLP*, 254–263.

Song, H.; Kim, M.; Park, D.; Shin, Y.; and Lee, J.-G. 2023. Learning From Noisy Labels With Deep Neural Networks: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 34(11): 8135–8153.

Sun, Z.; Zhang, Q.; Li, Y.; and Tan, Y. 2018. DPPDL: A Dynamic Partial-Parallel Data Layout for Green Video Surveillance Storage. *IEEE TCSVT*, 28(1): 193–205.

Tian, T.; Zhu, J.; and Qiaoben, Y. 2019. Max-Margin Majority Voting for Learning from Crowds. *IEEE TPAMI*, 41(10): 2480–2494.

Wang, G.; Liu, X.; Li, C.; Xu, Z.; Ruan, J.; Zhu, H.; Meng, T.; Li, K.; Huang, N.; and Zhang, S. 2020. A Noise-Robust Framework for Automatic Segmentation of COVID-19 Pneumonia Lesions From CT Images. *IEEE TMI*, 39(8): 2653–2663.

Wang, Z.; Hu, G.; and Hu, Q. 2020. Training Noise-Robust Deep Neural Networks via Meta-Learning. In *CVPR*, 4523–4532.

Wazny, K. 2017. "Crowdsourcing" ten years in: A review. *Journal of Global Health*, 7(2).

Wei, H.; Xie, R.; Feng, L.; Han, B.; and An, B. 2022a. Deep Learning From Multiple Noisy Annotators as A Union. *IEEE TNNLS*, 1–11.

Wei, J.; Zhu, Z.; Cheng, H.; Liu, T.; Niu, G.; and Liu, Y. 2022b. Learning with Noisy Labels Revisited: A Study Using Real-World Human Annotations. In *ICLR*.

Whitehill, J.; Ruvolo, P.; Wu, T.; Bergsma, J.; and Movellan, J. R. 2009. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *NeurIPS*, 2035–2043.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv*.

Xu, Y.; Zhu, L.; Jiang, L.; and Yang, Y. 2021. Faster Meta Update Strategy for Noise-Robust Deep Learning. In *CVPR*, 144–153.

Yu, D.; Deng, L.; and Dahl, G. 2010. Roles of pre-training and fine-tuning in context-dependent DBN-HMMs for real-world speech recognition. In *NeurIPS*. sn.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2021. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3): 107–115.

Zhang, H.; Li, S.; Wang, P.; Zeng, D.; and Ge, S. 2024. M3D: Dataset Condensation by Minimizing Maximum Mean Discrepancy. *AAAI*.

Zhang, J. 2022. Knowledge learning with crowdsourcing: a brief review and systematic perspective. *IEEE/CAA JAS*, 9(5): 749–762.

Zhang, Z.; and Pfister, T. 2021. Learning Fast Sample Reweighting Without Reward Data. In *ICCV*, 705–714.

Zheng, G.; Awadallah, A. H.; and Dumais, S. T. 2021. Meta Label Correction for Noisy Label Learning. In *AAAI*, 11053–11061.

Zheng, Y.; Li, G.; Li, Y.; Shan, C.; and Cheng, R. 2017. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5): 541–552.

Zoph, B.; and Le, Q. V. 2017. Neural Architecture Search with Reinforcement Learning. In *ICLR*.