Deep Learning From Multiple Noisy Annotators as A Union

Hongxin Wei, Renchunzi Xie, Lei Feng, Bo Han, and Bo An

***Abstract*— Crowdsourcing is a popular solution for large-scale data annotations. So far, various end-to-end deep learning methods have been proposed to improve the practical performance of learning from crowds. Despite their practical effectiveness, most of them have two major limitations—they do not hold learning consistency and suffer from computational inefficiency. In this article, we propose a novel method named UnionNet, which is not only theoretically consistent but also experimentally effective and efficient. Specifically, unlike existing methods that either fit a given label from each annotator independently or fuse all the labels into a reliable one, we concatenate the one-hot encoded vectors of crowdsourced labels provided by all the annotators, which takes all the labeling information as a union and coordinates multiple annotators. In this way, we can directly train an end-to-end deep neural network by maximizing the likelihood of this union with only a parametric transition matrix. We theoretically prove the learning consistency and experimentally show the effectiveness and efficiency of our proposed method.**

***Index Terms*— Annotators, crowdsourcing, noisy labels, transition matrix.**

I. Introduction

DEEP neural networks (DNNs) have achieved remarkable success on various real-world applications over the past years, while they heavily rely on a large number of training examples with accurate labels. To alleviate this issue, crowdsourcing provides a potential solution for large-scale annotations, which aims to elicit the correct label from crowdsourced labels. However, the collected crowdsourced labels could be very noisy due to human mistakes, especially for some difficult tasks like image annotation [1]–[3] and music genre classification (MGC) [4].

For training a classifier with noisy crowdsourced labels from multiple annotators, the key is how to abstract information from the imperfect crowdsourced labels. A traditional solution is to select the true label from crowdsourced labels by majority voting [5], [6] and train a classifier with the selected true label. However, this naive method would cause biased results because it simply assumes that all annotators are equally reliable, which is usually impractical since different annotators normally have different levels of expertise [7]. In order to capture the expertise levels of different annotators, some variants of expectation-maximization (EM) algorithm [7] were adapted to infer the ground-truth label and train the classifier alternately. Due to the iterative manner, EM-style algorithms are computationally expensive, especially when DNNs are used [2]. To improve the efficiency of training with DNNs, recent studies (e.g., CrowdLayer [8] and SpeeLFC [9]) aim to train an end-to-end deep neural network with multiple parametric annotator-specific transition matrices. Although they have achieved satisfactory practical performance, they still suffer from computational inefficiency and do not hold learning consistency.[1]

To further address the above two limitations, we propose a novel method named UnionNet, which is not only theoretically guaranteed but also experimentally effective and efficient. Specifically, unlike existing methods that either fit a given label from each annotator independently or fuse all the labels into a reliable one, we concatenate the one-hot encoded vectors of crowdsourced labels provided by all the annotators, which takes all the labeling information as a union and keeps it intact and coordinates multiple annotators. In this way, we can directly train an end-to-end deep neural network by maximizing the likelihood of the intact labeling information with only a transition matrix. In summary, the key contributions of this article are as follows.

1) We propose UnionNet, a novel method that takes the labeling information provided by all annotators as a union, and trains an end-to-end DNN maximizing the likelihood of this union with only a parametric transition matrix.
2) For theoretical guarantee, we prove the learning consistency of UnionNet by establishing an estimation error bound, which shows that obtained empirical risk

Manuscript received June 12, 2021; revised December 8, 2021; accepted April 8, 2022. This work was supported by the National Research Foundation, Singapore, through its Industry Alignment Fund–Pre-Positioning (IAF-PP) Funding Initiative. The work of Lei Feng was supported in part by the National Natural Science Foundation of China under Grant 62106028 and in part by the CAAI-Huawei MindSpore Open Fund. The work of Bo Han was supported in part by the NSFC Young Scientists Fund under Grant 62006202 and in part by the RGC Early Career Scheme under Grant 22200720. *(Corresponding author: Lei Feng.)*

Hongxin Wei, Renchunzi Xie, and Bo An are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: hongxin001@e.ntu.edu.sg; renchunzi.xie@ntu.edu.sg; boan@ntu.edu.sg).

Lei Feng is with the College of Computer Science, Chongqing University, Chongqing 400044, China (e-mail: lfeng@cqu.edu.cn).

Bo Han is with the Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China (e-mail: bhanml@comp.hkbu.edu.hk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3168696.

Digital Object Identifier 10.1109/TNNLS.2022.3168696

[1] Learning is consistent, if and only if the risk of the learned classifier converges to the risk of the optimal classifier, as the amount of training data approaches infinity, where the optimality is defined over a given hypothesis class.

minimizer by UnionNet would converge to the true risk minimizer as the amount of training data tends to infinity.

3) For practical performance, we conduct extensive experiments on synthesized as well as real-world crowdsourced datasets, and the experimental results demonstrate that UnionNet is superior to the state-of-the-art counterparts.
4) For training efficiency, we experimentally show that the training time of UnionNet is nearly the same as that of standard DNN trained with correct labels, which is significantly less than that of other end-to-end methods.

II. Related Work

In this section, we briefly review existing works on learning with crowdsourced labels. Most of the existing methods belong to two categories: EM-style algorithms and end-to-end algorithms.

A. EM-Style Algorithms

For learning with crowdsourced labels, a large number of methods [7], [10] are based on the key idea of the EM algorithm [11]. For example, GLAD [10] proposed to first infer the ground-truth label for each instance by aggregating the given labels from multiple annotators, and then train the classifier based on the inferred labels. Another important work in this direction [7] considers the ground-truth labels as latent variables, thereby jointly learning different annotators' expertise and training an effective logistic regression classifier. This prominent work inspired a number of follow-up works, such as Gaussian process classifiers [12], supervised latent Dirichlet allocation [13], and convolutional neural networks with softmax outputs [2], [14].

B. End-to-End Algorithms

Despite the effectiveness of the above EM-style algorithms, their computational complexity of training could be high due to the alternating optimization manner. This issue may deteriorate when DNNs are used for training. To ease this problem, an end-to-end approach called CrowdLayer [8] for deep learning with crowdsourced labels was proposed, which is able to directly train a DNN with crowdsourced labels through a crowd layer that models the annotator-specific transition matrix of each annotator. In this way, the classifier and the crowd layer can be trained simultaneously by backpropagation [15], thereby improving the training efficiency. Following this idea, a recent method called SpeeLFC [9] proposed a probabilistic model that learns an interpretable transition matrix for each annotator, which also allows an end-to-end structure for training a DNN. In addition, there exist some end-to-end methods that do not rely on any annotator-specific transition matrix. For example, DoctorNet [1] aims to train DNNs that exploit different annotators' information with different softmax output layers. Its variant WDN [1] trains the classifier with multiple output layers and learns combination weights for aggregating the outputs.

TABLE I

Comparisons Between Our Proposed Algorithms and Related Studies. "End-to-End": Does the Algorithm Follow an End-to-End Learning Architecture? "One-Way": Does the Algorithm Train the Deep Neural Network Without Multiple Transition Matrices (for Improving Training Efficiency)? "Theory": Is the Algorithm Theoretically Guaranteed?

	End-to-End	One-Way	Theory
MajorVote [5]	✗	✓	✓
CrowdLayer [8]	✓	✗	✗
DoctorNet [1]	✓	✗	✗
WDN [1]	✓	✗	✗
SpeeLFC [9]	✓	✗	✗
UnionNet-A (Ours)	✓	✓	✗
UnionNet-B (Ours)	✓	✓	✓

Compared with EM-style algorithms, these end-to-end algorithms that accommodate DNNs achieve comparable even better performance. However, these algorithms still have two major limitations. They do not hold learning consistency and do not have high training efficiency due to the existence of multiple annotator-specific matrices. In the following sections, we will first introduce some preliminaries of this article, and then present a novel end-to-end method (with two practical algorithms) that relies on only a single transition matrix, which not only addresses the above two limitations, but also achieves better performance than other counterparts. The detailed comparisons between our proposed algorithms and related studies are presented in Table I.

The outline of this article is as follows. In Section III, we present the problem setting and the formulation of CrowdLayer. In Section IV, we introduce the two solutions of UnoinNet and provide theoretical analysis in Section V. In Section VI, we experimentally analyze the proposed methods on synthesized datasets and real-world datasets. Moreover, we also provide ablation study and training efficiency analysis to further demonstrate the advantage of our methods. Finally, we conclude the article in Section VII.

III. Preliminaries

In this section, we introduce the problem setting of learning with crowdsourced labels and the formulation of CrowdLayer, which is a representative related work.

A. Problem Setting

Let $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^{n}$ be a crowdsourced dataset that includes n examples with k classes, where for each instance $\boldsymbol{x}_i \in \mathbb{R}^d$, we receive a set of crowdsourced labels $Y_i = \{\tilde{y}_i^j \mid j = 1, \ldots, m\}$, with $\tilde{y}_i^j$ representing the label provided by the jth annotator in a set of m annotators. It is worth noting that the number of crowdsourced labels of different examples could be different. Hence, $|Y_i|$ may not be equal to $|Y_j|$ if $i \neq j$. Following [7]–[9], [16], we also assume each instance $\boldsymbol{x}_i$

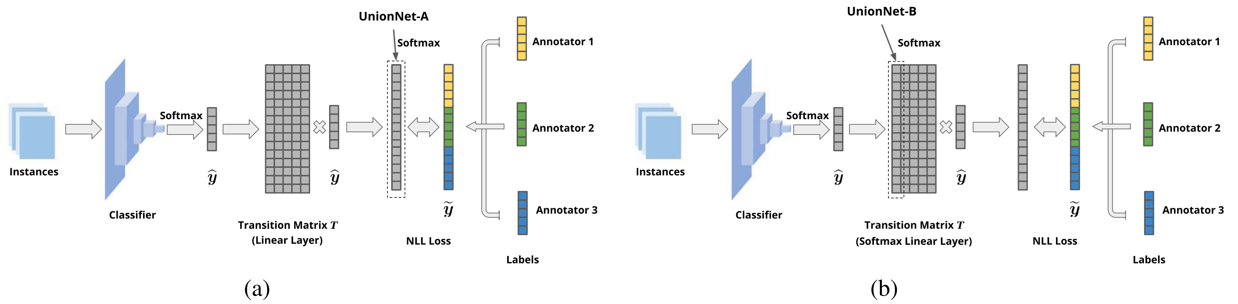


Fig. 1. Training schemes of UnionNet (ours) for the classification task with five classes and three annotators. For the two algorithms of UnionNet, softmax operation is either applied on transformed outputs for (a) UnionNet-A or columns of the transition matrix for (b) UnionNet-B. In the test stage, $\hat{\boldsymbol{y}}$ can be readily used to make predictions for an unseen instance $\boldsymbol{x}_i$.

has its corresponding latent ground-truth label y_i and it is related to the crowdsourced labels Y_i. It is worth noting that learning with crowdsourced data is a multiclass classification problem, which means there is only a true label for each training instance. This setting is different from multilabel learning [17]–[21] where multiple true labels are provided for each training instance. Under this setting, our goal is to train a classifier f based on a crowdsourced dataset $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^{n}$ that could accurately predict the ground-truth label of an unseen instance in the test phase.

B. Formulation of CrowdLayer

Instead of using an alternating optimization manner of EM-style algorithms, CrowdLayer [8] aims to train an end-to-end DNN that simultaneously optimizes all the components. Specifically, CrowdLayer relies on a crowd layer that takes the softmax outputs of a standard DNN model as the input, and learns multiple parametric annotator-specific transition matrices to fit the given label of each annotator independently.

For each annotator j, we define an annotator-specific crowd layer by a transition matrix $\boldsymbol{T}^j$, which is equivalent to a linear layer without bias, i.e., $\boldsymbol{T}^j\hat{\boldsymbol{y}}$. In CrowdLayer, a transition matrix is explicitly learned for each annotator, thereby enabling crowdsourced labels to propagate errors through the whole network structure. Formally, let $\hat{\boldsymbol{y}}$ be the softmax output of a standard classifier. Given a loss function $\mathcal{L}$ (e.g., categorical cross-entropy loss) for classification, CrowdLayer minimizes the following objective:

$$\min_{\boldsymbol{\Theta}} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\mathcal{L}\big(\mathrm{softmax}(\boldsymbol{T}^j\hat{\boldsymbol{y}}_i), \tilde{y}_i^j\big) \qquad (1)$$

where $\boldsymbol{\Theta} = \{\{\boldsymbol{T}^j\}_{j=1}^{m}, f\}$ denotes the set of all learning parameters that include the parameters of transition matrices and the classifier. Trained with the above objective function, CrowdLayer achieves the goal of directly learning from crowdsourced labels in an end-to-end manner. However, CrowdLayer does not hold learning consistency. In addition, most existing end-to-end methods, including CrowdLayer and SpeeLFC, need to forward propagate through multiple linear layers in a for-loop way. It also introduces extra computational complexity, especially when scaling to a large number of annotators.

IV. Proposed UnionNet

As described above, the training inefficiency of existing end-to-end algorithms is caused by the for-loop forward propagation of multiple independent annotator-specific transition matrices. Hence, we conjecture that if we use a single transition matrix, the training efficiency may be improved.

A. Union of Crowdsourced Labels

We present a novel end-to-end method called UnionNet, which allows us to leverage a single transition matrix to directly train a DNN with the crowdsourced labels provided by multiple annotators. UnionNet not only experimentally outperforms other end-to-end counterparts, but also holds learning consistency and has high training efficiency. As shown in Fig. 1(b), instead of learning an independent annotator-specific transition matrix for each annotator, we propose to learn a single transition matrix. In this way, we could transform the softmax outputs of the standard classifier via this transition matrix to fit a union of labeling information provided by all the annotators.

By taking the labeling information provided by all the annotators as a union, our proposed UnionNet is advantageous in two aspects. First, unlike previous methods [8], [9] that treat each annotator equally, UnionNet naturally coordinates the contributions of different annotators on the true label. Specifically, in CrowdLayer, each annotator would contribute a loss independently, while in our UnionNet, there is only a single loss that is contributed by all the annotators. Therefore, UnionNet takes into account the collaboration of all the annotators, while previous methods treat each annotator independently. Hence, UnionNet is expected to achieve better performance, especially in the correlated settings. Second, previous methods conduct forward propagation through multiple linear layers (i.e., transition matrices) in a for-loop way. In contrast, UnionNet only has a single transition matrix, hence it avoids that for-loop way. Therefore, UnionNet would be more computationally efficient.

Therefore, the question becomes how to combine the crowdsourced labels of all the annotators as a union. Our proposed UnionNet provides a simple yet effective answer to this question. Concretely, we can concatenate the vectors of one-hot encoded crowdsourced labels provided by different annotators to form a united vector $\widetilde{\boldsymbol{y}}_i$

$$\widetilde{\boldsymbol{y}}_i = \text{concatenate}\left(\boldsymbol{e}^{(\tilde{y}_i^1)}, \boldsymbol{e}^{(\tilde{y}_i^2)}, \ldots, \boldsymbol{e}^{(\tilde{y}_i^m)}\right) \tag{2}$$

where $\boldsymbol{e}^{(\tilde{y}_i^j)} := (0, \ldots, 1, \ldots, 0)^\top \in \{0, 1\}^k$ is a one-hot vector and only the $\tilde{y}_i^j$th entry of $\boldsymbol{e}^{(\tilde{y}_i^j)}$ is 1. When the jth annotator does not provide a label for the ith instance $\boldsymbol{x}_i$, we define the vector $\boldsymbol{e}^{(\tilde{y}_i^j)}$ as a zero vector. In this way, all the crowdsourced labels are processed as a union. Here, one may think that we could use another combination method to produce a union of crowdsourced labels. For example, we could sum up the vectors of one-hot encoded crowdsourced labels together to create a single union vector. However, this simple addition operation could not fully capture the complex relationships between the crowdsourced labels and the ground-truth label. In contrast, our proposed concatenation operation in (2) enables the union of crowdsourced labels to contain the information of annotators so that it can automatically coordinate the influences of different annotators. We also experimentally demonstrate the advantage of our concatenation operation over the addition operation by conducting ablation studies in the experiment section.

B. Training Objective

By taking the crowdsourced labels as a union vector $\tilde{\boldsymbol{y}}_i$, our final goal is to train an end-to-end deep model by fitting this union vector $\tilde{\boldsymbol{y}}_i$. To achieve this goal, we adopt the widely used maximum likelihood principle [22], [23]. Formally speaking, we would like to maximize the $p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x})$, where $p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x})$ can be formulated as

$$\begin{aligned} p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x}) = \sum_y p(\widetilde{\boldsymbol{y}}, y \mid \boldsymbol{x}) &= \sum_y p(\widetilde{\boldsymbol{y}} \mid y, \boldsymbol{x}) p(y \mid \boldsymbol{x}) \\ &= \sum_y p(\widetilde{\boldsymbol{y}} \mid y) p(y \mid \boldsymbol{x}) \end{aligned} \tag{3}$$

where the last equality holds due to the widely used assumption [7]–[9] that the crowdsourced labels are only related to the ground-truth label y and independent of the instance $\boldsymbol{x}$. It is worth noting that there is a similar assumption in the area of noisy-label learning [24]–[37] that the observed label noise is class-conditional. In this way, the expected risk of our proposed UnionNet could be formulated as the negative expected log-likelihood

$$R(f) = -\mathbb{E}_{p(\boldsymbol{x}, \widetilde{\boldsymbol{y}})}\left[\log p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x})\right] \tag{4}$$

where $p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x})$ denotes the distribution of crowdsourced data. In terms of (3), the key of our proposed end-to-end learning method is how to empirically approximate the union vector $\widetilde{\boldsymbol{y}}_i$ by using different components in our deep architecture. Here, we provide two solutions, which are named UnionNet-A and UnionNet-B, respectively.

1) UnionNet-A: Motivated by the training objective of CrowdLayer [8] in (1), we could empirically approximate the union vector $\widetilde{\boldsymbol{y}}_i$ of $\boldsymbol{x}_i$ by $\text{softmax}(\boldsymbol{T}\widehat{\boldsymbol{y}}_i)$ where $\widehat{\boldsymbol{y}}_i$ denotes the softmax outputs of the standard classifier and $\boldsymbol{T} \in \mathbb{R}^{km} \times \mathbb{R}^k$ is a parametric transition matrix, which is defined as a linear layer without bias. It is worth noting that we need to further conduct the softmax operation on $\boldsymbol{T}\widehat{\boldsymbol{y}}_i$ to make it always nonnegative, as there is no constraints posed on the transition matrix $\boldsymbol{T}$. Guided by the expected risk in (4), we have the following training objective of UnionNet-A:

$$\min_{\boldsymbol{\Theta}} \; -\frac{1}{n}\sum_{i=1}^n \widetilde{\boldsymbol{y}}_i \log\left(\text{softmax}(\boldsymbol{T}\widehat{\boldsymbol{y}}_i)\right) \tag{5}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{T}, f\}$ denotes the set of all learning parameters. It is worth noting that for UnionNet-A, the parametric transition matrix $\boldsymbol{T}$ is not able to model the conditional probability $p(\widetilde{\boldsymbol{y}} \mid y)$ since the elements of $\boldsymbol{T}$ may be smaller than 0. As a consequence, UnionNet-A may not hold any theoretical guarantee, since it does strictly follow from (3) to (4). To alleviate this issue, we further present the other solution UnionNet-B, which is theoretically guaranteed.

2) UnionNet-B: Motivated by SpeeLFC [9] that employs a probabilistic parametric transition matrix to enhance the interpretability for each annotator, we could also empirically approximate the union vector $\widetilde{\boldsymbol{y}}_i$ of $\boldsymbol{x}_i$ by $\text{softmax}(\boldsymbol{T})\widehat{\boldsymbol{y}}_i$ where $\text{softmax}(\boldsymbol{T})$ means taking the softmax operation on each column of the parametric transition matrix $\boldsymbol{T}$, which is a linear layer without bias. It is worth noting that each column of $\text{softmax}(\boldsymbol{T})$ indicates that the probability vector of each true label y becomes a specific union vector $\widetilde{\boldsymbol{y}}$. Therefore, we need to impose the softmax operation on each column of $\boldsymbol{T}$ to satisfy the requirements of $p(\widetilde{\boldsymbol{y}} \mid y)$, i.e., $\sum_{\widetilde{\boldsymbol{y}}} p(\widetilde{\boldsymbol{y}} \mid y) = 1$ and $p(\widetilde{\boldsymbol{y}} \mid y) \geq 0$ for any $\widetilde{\boldsymbol{y}}$ and y. Other weakly supervised learning settings have also provided various attempts to estimate such kind of probabilistic transition matrix using anchor points [24], [38], or learn it as a noise adaptation layer [39]. Our proposed UnionNet-B adopts the latter strategy, and the training objective of UnionNet-B is expressed as

$$\min_{\boldsymbol{\Theta}} \; -\frac{1}{n}\sum_{i=1}^n \widetilde{\boldsymbol{y}}_i \log\left(\text{softmax}(\boldsymbol{T})\widehat{\boldsymbol{y}}_i\right) \tag{6}$$

where $\boldsymbol{\Theta} = \{\boldsymbol{T}, f\}$ denotes the set of all learning parameters. It is worth noting that SpeeLFC employs multiple probabilistic parametric transition matrices, each for an annotator, and it does not hold any guarantee. In contrast, our proposed UnionNet-B only uses a single probabilistic parametric transition matrix, and we will show that it is theoretically consistent and has higher training efficiency.

3) Differences Between UnionNet-A and UnionNet-B: As shown in Fig. 1, both the two proposed methods optimize a single probabilistic parametric transition matrix. Differently, UnionNet-A applies the softmax operation on transformed outputs while UnionNet-B uses it on columns of the transition matrix. As a result, UnionNet-B is theoretically consistent but UnionNet-A do not hold any theoretical guarantee. In addition, the softmax operation on columns of the parametric transition matrix provides a better way to automatically coordinate the influences of different annotators, which is clearly supported by the empirical results in Table III. With such a guarantee, UnionNet-B is expected to achieve better empirical performance than UnionNet-A.

Algorithm 1 UnionNet

```
Input: Classifier $f$ with $\boldsymbol{W}$, transition matrix $\boldsymbol{T}$, number of epochs $E_{\max}$;
for $e = 1, 2, \dots, E_{\max}$ do
    $\widehat{\boldsymbol{y}} = f(\boldsymbol{x}, \boldsymbol{W})$;
    Obtain union $\widetilde{\mathbf{y}}$ by (2);
    Obtain $\mathcal{L}$ by (5) for UnionNet-A or (6) for UnionNet-B based on $\widetilde{\boldsymbol{y}}$ and $\widehat{\boldsymbol{y}}$;
    Update $\{\boldsymbol{W}, \boldsymbol{T}\}$ by $\mathcal{L}$ using Adam optimization method.
end for
Output: $\boldsymbol{W}$
```

C. Initialization Strategy

A good initialization strategy for the parametric transition matrix $\boldsymbol{T}$ is helpful for the convergence of the whole network architecture in the training stage. At the beginning of training, we could treat the linear layer (which represents the whole transition matrix) as a simple concatenation of m sub-layers and each of them represents a transition matrix of a annotator: $\boldsymbol{T} = \text{concatenate}(\boldsymbol{T}^1, \boldsymbol{T}^2, \dots, \boldsymbol{T}^m)$. We adopt the following two initialization strategies in the experiments.

For the first initialization strategy, we follow Crowd-Layer [8] to initialize the sub-layer for each annotator as:

$$\boldsymbol{T}^r(a, b) = (1 - \epsilon)\mathbb{I}_{\{a=b\}} + \frac{\epsilon}{C - 1}\mathbb{I}_{\{a \neq b\}} \tag{7}$$

where $\mathbb{I}_{\text{condition}} = 1$ when condition is true, $\mathbb{I}_{\text{condition}} = 0$ when condition is false, and ϵ denotes an extremely small constant, which is fixed at 10^{-5} in our experiments.

For the second initialization strategy, we follow [7], [14] to initialize the sub-layer for each annotator as:

$$\boldsymbol{T}^r(a, b) = \log \frac{\sum_{i=1}^{n} Q(y_i = a)\mathbb{I}_{\{\tilde{y}_i^j = b\}}}{\sum_{i=1}^{n} Q(y_i = a)} \tag{8}$$

where $Q(y_i = a) := (1/m) \sum_{r=1}^{m} \mathbb{I}_{\{\tilde{y}_i^j = a\}}$.

We will provide detailed information about which strategy will be adopted on which dataset in the experiments. The details of our proposed UnionNet are shown in Algorithm 1. Specifically, UnionNet is more computationally efficient since it only optimizes a single transition matrix during training, thereby avoiding the for-loop way in previous methods. Moreover, UnionNet explores the relationships between annotators by the concatenate operation, thereby being able to achieve better practical performance.

V. Theoretical Analysis

For UnionNet, the crowdsourced labels for each instance are taken as a union vector. In this way, we could represent the provided crowdsourced datasets as $\{\boldsymbol{x}_i, \widetilde{\boldsymbol{y}}_i\}_{i=1}^{n}$, where each example is assumed to be independently sampled from the distribution of crowdsourced data $p(\boldsymbol{x}, \widetilde{\boldsymbol{y}})$. It is noteworthy that for UnionNet-B, $\text{softmax}(\boldsymbol{T})$ could perfectly model the conditional probability $p(\widetilde{\boldsymbol{y}} \mid y)$ in (3). Hence, we only focus on the theoretical analysis of UnionNet-B, and represent the used transition matrix $\text{softmax}(\boldsymbol{T})$ as $\widetilde{\boldsymbol{T}}$.

We define the model class as $\mathcal{F} \subset \{f : \mathbb{R}^d \mapsto \mathbb{R}^k\}$ and define the true risk minimizer as

$$f^\star = \underset{f \in \mathcal{F}}{\arg\min}\ \mathbb{E}_{p(\boldsymbol{x}, y)}\big[\mathcal{L}(f(\boldsymbol{x}), y)\big] \tag{9}$$

where $p(\boldsymbol{x}, y)$ denotes the underlying data distribution of the normal example $(\boldsymbol{x}, y)$ and y is the true label of $\boldsymbol{x}$. Then we define the obtained empirical risk minimizer by minimizing (6) as $\widehat{f}$. In the following, we will prove that the obtained empirical risk minimizer $\widehat{f}$ would converge to the true risk miminizer $f^\star$ as the collected crowdsourced training data approaches infinity.

First of all, we prove that the obtained expected risk minimizer $\widetilde{f}^\star$ by minimizing (4) is exactly the same as the true risk minimizer $f^\star$ if the categorical cross entropy loss is used in (9).

Lemma 1: Suppose $g(\boldsymbol{x}) = \text{softmax}(f(\boldsymbol{x}))$, by optimizing (9) with categorical cross entropy loss, the optimal mapping $g^\star$ (w.r.t. $f^\star$) satisfies $g_i^\star(\boldsymbol{x}) = p(y = i \mid x), \forall i \in [k]$.

Proof: If the cross entropy loss is used, we have the following optimization problem:

$$\phi(g) = -\sum_{i=1}^{k} p(y = i \mid \boldsymbol{x}) \log(g_i(\boldsymbol{x}))$$
$$\text{s.t.} \sum_{i=1}^{k} g_i(\boldsymbol{x}) = 1.$$

By using the Lagrange multiplier method, we can obtain the following nonconstrained optimization problem:

$$\Phi(g) = -\sum_{i=1}^{k} p(y = i \mid \boldsymbol{x}) \log(g_i(\boldsymbol{x})) + \lambda(\sum_{i=1}^{k} g_i(\boldsymbol{x}) - 1)).$$

By setting the derivative to 0, we obtain

$$g_i^\star(\boldsymbol{x}) = \frac{1}{\lambda} p(y = i \mid \boldsymbol{x}).$$

Because $\sum_{i=1}^{k} g_i^\star(\boldsymbol{x}) = 1$ and $\sum_{i=1}^{k} p(y = i \mid \boldsymbol{x}) = 1$, we have

$$\sum_{i=1}^{k} g_i^\star(\boldsymbol{x}) = \frac{1}{\lambda} \sum_{i=1}^{k} p(y = i \mid \boldsymbol{x}) = 1.$$

Therefore, we can easily obtain $\lambda = 1$. In this way, $g_i^\star = (1/\lambda) p(y = i \mid \boldsymbol{x}) = p(y = i \mid \boldsymbol{x})$, which concludes the proof of Lemma 1. □

Theorem 1: When the transition matrix $\widetilde{\boldsymbol{T}}$ has full rank and the condition in Lemma 1 is satisfied, our expected risk minimizer $\widetilde{f}^\star$ is the same as the true risk minimizer $f^\star$, i.e., $\widetilde{f}^\star = f^\star$.

Proof: According to Lemma 1, when we obtain $\widetilde{f}^\star$ by optimizing (4), we could optimally fit the conditional probability $p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x})$

$$q^\star(\boldsymbol{x}) = p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x}).$$

Let us introduce $v(\boldsymbol{x}) = p(y \mid \boldsymbol{x})$ and $\widetilde{v}(\boldsymbol{x}) = p(\widetilde{\boldsymbol{y}} \mid \boldsymbol{x})$. According to (3), we have

$$\widetilde{v}(\boldsymbol{x}) = \widetilde{\boldsymbol{T}} v(\boldsymbol{x}).$$

Since $q^\star(\boldsymbol{x}) = \widetilde{v}(\boldsymbol{x})$, we have $q^\star(\boldsymbol{x}) = \widetilde{\boldsymbol{T}}v(\boldsymbol{x})$. Besides, we obtain $\widetilde{g}^\star(\boldsymbol{x})$ $(\widetilde{g}^\star(\boldsymbol{x}) = \mathrm{softmax}(\widetilde{f}^\star(\boldsymbol{x})))$ by optimizing (4). Thus $q^\star(\boldsymbol{x}) = \widetilde{\boldsymbol{T}}\widetilde{g}^\star(\boldsymbol{x})$, which further ensures that $\widetilde{\boldsymbol{T}}v(\boldsymbol{x}) = \widetilde{\boldsymbol{T}}\widetilde{g}^\star(\boldsymbol{x})$. In this way, if the transition matrix $\widetilde{\boldsymbol{T}}$ has full rank, we can obtain $v(\boldsymbol{x}) = \widetilde{g}^\star(\boldsymbol{x})$. As we have shown that $v(\boldsymbol{x}) = p(y \mid \boldsymbol{x}) = g^\star(\boldsymbol{x})$ where $g^\star(\boldsymbol{x}) = \mathrm{softmax}(f^\star(\boldsymbol{x}))$, we can obtain $\widetilde{g}^\star(\boldsymbol{x}) = g^\star(\boldsymbol{x})$, which implies that $\widetilde{f}^\star = f^\star$. Thus Theorem 1 is proven. □

Next, we theoretically establish an estimation error bound, which demonstrates the learning consistency of UnionNet-B based on Rademacher complexity [40]. From (6), we can introduce a composite loss $\widetilde{\mathcal{L}}(f(\boldsymbol{x}), \widetilde{\boldsymbol{y}}) = -\widetilde{\boldsymbol{y}}\log(\widetilde{\boldsymbol{T}}\mathrm{softmax}(f(\boldsymbol{x})))$. Let $\mathcal{F}$ be represented as $\mathcal{F} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_k$, where $\mathcal{F}_i := \{f : \boldsymbol{x} \mapsto f_i(\boldsymbol{x}) \mid f \in \mathcal{F}\}, \forall i \in [k]$. We denote by $\mathfrak{R}_n(\mathcal{F}_y)$ the Rademacher complexity of $\mathcal{F}_y$ given the sample size n over $p(\boldsymbol{x})$. Then we have the following theorem.

Theorem 2: Assume $\widetilde{\mathcal{L}}$ is ρ-Lipschitz $(\rho < 0 < \infty)$ continuous w.r.t. $f(\boldsymbol{x})$ and upper bounded by M, i.e., for any $(\boldsymbol{x}, \widetilde{\boldsymbol{y}}) \sim p(\boldsymbol{x}, \widetilde{\boldsymbol{y}})$, $\widetilde{\mathcal{L}}(f(\boldsymbol{x}), \widetilde{\boldsymbol{y}}) \leq M$. Then, for any $\delta > 0$, with probability at least $1 - \delta$

$$R(\widehat{f}) - R(f^\star) \leq 2\sqrt{2}\rho\sum_{y=1}^{k}\mathfrak{R}_n(\mathcal{F}_y) + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}. \tag{10}$$

Proof: Before proving Theorem 2, we first introduce the following notations: $\widetilde{f}^\star = \arg\min_{f\in\mathcal{F}} R(f)$ and $\widehat{f} = \arg\min_{f\in\mathcal{F}} \widehat{R}(f)$, and $\widehat{R}(f)$ represents the objective in (6) of UnionNet-B, which is an empirical approximation of $R(f)$. Then, the following inequality holds:

$$\begin{aligned} R(\widehat{f}) - R(f^\star) &\leq R(\widehat{f}) - \widehat{R}(\widehat{f}) + \widehat{R}(\widehat{f}) - R(\widetilde{f}^\star) \\ &\leq R(\widehat{f}) - \widehat{R}(\widehat{f}) + R(\widehat{f}) - R(\widetilde{f}^\star) \\ &\leq 2\sup_{f\in\mathcal{F}}\left|R(f) - \widehat{R}(f)\right|. \end{aligned}$$

Then, we introduce the uniform deviation bound, which is useful to derive estimation error bounds. The proof can be found in some textbooks such as [40, Th. 3.1].

Lemma 2: Let Z be a random variable drawn from a probability distribution with density $\mu, \mathcal{H} = \{h : \mathcal{Z} \mapsto [0, M]\}$ $(M > 0)$ be a class of measurable functions, $\{z_i\}_{i=1}^n$ be i.i.d. examples drawn from the distribution with density μ. Then, for any, delta > 0, with probability at least $1 - \delta$

$$\sup_{h\in\mathcal{H}}\left|\mathbb{E}_{Z\sim\mu}\big[h(Z)\big] - \frac{1}{n}\sum_{i=1}^{n}h(z_i)\right| \leq 2\mathfrak{R}_n(\mathcal{H}) + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}.$$

Based on Lemma 2, we can directly obtain the following lemma.

Lemma 3: Assume $\widetilde{\mathcal{L}}$ is upper bounded by M, i.e., for any $(\boldsymbol{x}, \widetilde{\boldsymbol{y}}) \sim p(\boldsymbol{x}, \widetilde{\boldsymbol{y}})$, $\widetilde{\mathcal{L}}(f(\boldsymbol{x}), \widetilde{\boldsymbol{y}}) \leq M$. Then, for any $\delta > 0$, with probability at least $1 - \delta$

$$R(\widehat{f}) - R(f^\star) \leq 2\mathfrak{R}_n(\widetilde{\mathcal{L}} \circ \mathcal{F}) + M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}$$

where $\widetilde{\mathcal{L}} \circ \mathcal{F}$ is defined as $\widetilde{\mathcal{L}} \circ \mathcal{F} = \{\widetilde{\mathcal{L}} \circ f \mid f \in \mathcal{F}\}$.

Then, we need to upper bound $\mathfrak{R}_n(\widetilde{\mathcal{L}} \circ \mathcal{F})$. Since $\widetilde{\mathcal{L}}$ is assumed to be ρ-Lipschitz continuous w.r.t. $f(\boldsymbol{x})$, according to the Rademacher vector contraction inequality [41], we have

$$\widehat{\mathfrak{R}}_n(\widetilde{\mathcal{L}} \circ \mathcal{F}) \leq \sqrt{2}\rho\sum_{y=1}^{k}\widehat{\mathfrak{R}}_n(\mathcal{F}_y)$$

where $\widehat{\mathfrak{R}}_n(\mathcal{F})$ and $\widehat{\mathfrak{R}}_n(\mathcal{F}_y)$ are the empirical Rademacher complexity [40] of $\mathcal{F}$ and $\mathcal{F}_y$. By taking the expectation over $p(\boldsymbol{x})$, we have $\mathfrak{R}_n(\widetilde{\mathcal{L}} \circ \mathcal{F}) \leq \sqrt{2}\rho\sum_{y=1}^{k}\mathfrak{R}_n(\mathcal{F}_y)$. By further taking into account Lemma 3, Theorem 2 is proven. □

Generally, $\mathfrak{R}_n(\mathcal{F}_y)$ can be bounded by $C_{\mathcal{F}}/\sqrt{n}$ for a positive constant $C_{\mathcal{F}}$ [28], [42], [43]. Hence, Theorem 2 demonstrates that the empirical risk minimizer $\widehat{f}$ (obtained by learning from only crowdsourcing data) converges to the true risk minimizer $f^\star$ (obtained by minimizing the expected classification risk on fully labeled data) as the number of crowdsourcing data approaches infinity $(n \to \infty)$. Such a theoretical result indicates that we can obtain a reasonable classifier by directly using our UnionNet-B with only crowdsourcing data, and such a classifier would be better if more crowdsourcing data are provided. In addition, we can observe that the convergence rate of the derived bound in Theorem 2 is $\mathcal{O}_p(1/\sqrt{n})$ where $\mathcal{O}_p$ denotes the order in probability. This order is known as the optimal parametric rate for empirical risk minimization without additional assumptions [44].

VI. Experiments

A. Benchmark Datasets

1) Synthesized Datasets: Four widely used synthesized datasets are used in our experiments [8], [14], [45]. Dogs versus Cats dataset consists of 25 000 images from two classes dogs and cats, which is split into a 12 500-image training set and a 12 500-image test set. CIFAR-10 dataset consists of 60 000 images from ten classes, which is split into a 50 000-image training set and a 10 000-image test set. MNIST dataset also contains 60 000 training images and 10 000 testing images. LUNA16 dataset consists of 888 CT scans for lung nodule. We preprocessed the CT scans by generating 8106 50×50 gray-scale images, which is split into a 6484-image training set and a 1622-image testing set.

For each synthesized dataset, we follow [14] to generate two groups of crowdsourced labels: labels provided by (H) annotators with relatively high expertise; (L) annotators with relatively low expertise. Besides, we consider two cases of mistakes: independent mistakes, where senior annotators are mutually conditionally independent, and correlated mistakes, where senior annotators are mutually conditional independent, and each junior annotators copies one of the senior annotators. For each of the situations (H) and (L), the two cases have the same senior annotators. For independent mistakes, we set the number of annotators to five and ten for the situations (H) and (L), respectively. For correlated mistakes, the number of senior annotators is set to five and ten for the situations (H) and (L), while the number of junior annotators is set to five and two for the situations (H) and (L). In the following, we will introduce the details of the two cases for different datasets.

2) Independent Mistakes: For Dogs versus Cats, in situation (H), some senior annotators are more familiar with cats, while others make better judgments on dogs. Specifically, the expertise of five annotators are A: 0.6/0.8, B: 0.6/0.6, C: 0.9/0.6, D: 0.7/0.7, E: 0.6/0.7. For example, the expertise of annotator A is 0.6/0.8 in the sense that if the ground truth is dog/cat, she labels the image as "dog"/"cat" with probability 0.6/0.8, respectively. In situation (L), all ten seniors' expertise are 0.55/0.55.

For CIFAR-10, in situation (H), we generate annotators who may make mistakes in distinguishing the hard pairs: cat/dog, deer/horse, airplane/bird, automobile/trunk, frog/ship, but can perfectly distinguish other easy pairs (e.g. cat/frog), which makes sense in practice. When they cannot distinguish the pair, some of them may label the pair randomly and some of them label the pair the same class. In detail, for each hard pair, annotator A label the pair the same class (e.g., A always labels the image as "cat" when the image has cats or dogs), annotator B labels the pair uniformly at random (e.g., B labels the image as "cat" with the probability 0.5 and "dog" with the probability 0.5 when the image has cats or dogs). Annotator C is familiar with mammals so she can distinguish cat/dog and deer/hose, while for other hard pairs, she label each of them uniformly at random. Annotator D is familiar with vehicles so she can distinguish airplane/bird, automobile/trunk, and frog/ship, while for other hard pairs, she always label each of them the same class. Annotator E does not have special expertise. For each hard pair, annotator E labels them correctly with the probability 0.6. In situation (L), all ten senior experts label each image correctly with probability 0.2 and label each image as other false classes uniformly with probability (0.8/9). We use the same settings for the MNIST dataset.

For LUNA16, in situation (H), some senior annotators tend to label the image as "benign" while others tend to label the image as "malignant." Their expertise for benign/malignant are: A: 0.6/0.9, B: 0.7/0.7, C: 0.9/0.6, D: 0.6/0.7, E: 0.7/0.6. In situation (L), all ten seniors' expertise are 0.6/0.6.

3) Correlated Mistakes: For Dogs versus. Cats, MNIST, CIFAR-10 and LUNA16, in situation (H), two junior annotators copy annotator A's labels and three junior annotators copy annotator C's labels; in situation (L), one junior annotator copies annotator A's labels and another junior annotator copies annotator C's labels.

4) Real-World Datasets: Two widely used real-world datasets are used in our experiments [8], [9]. LabelMe dataset consists of a total of 2688 images, where 1000 of them are used to obtain labels from multiple annotators from Amazon Mechanical Turk [46] and the remaining 1688 images are used for evaluation. Each image is labeled by an average of 2.547 workers, with a mean accuracy of 69.2% [47]. MGC dataset contains 700 examples (with crowdsourced annotations) of songs with 30 s in length and are divided into ten different music genres (classical, country, disco, etc.) [4].

5) Initialization: For the transition matrix in our algorithms, we use (8) for initialization on CIFAR10 and MGC. On other datasets, we initialize the layer by (7). The same setting is applied for CrowdLayer. For SpeeLFC [9], we use the setting described in their article: the values on the diagonal elements of each transition matrix are initially set to 4.7, and the other values are fixed at 1.

TABLE II

SUMMARY OF TRAINING SETTINGS FOR DIFFERENT DATASETS

	Batch Size	Learning rate	Epochs	Architecture
Dogs vs. Cats	16	10^{-4}	20	4-layer CNN
MNIST	64	10^{-5}	100	2-layer MLP
CIFAR-10	64	10^{-3}	50	VGG-16
LUNA16	16	10^{-4}	20	4-layer CNN
LabelMe	64	10^{-4}	20	VGG-16
MGC	100	10^{-3}	5000	Linear model

B. Compared Algorithms

We compare the two algorithms of UnionNet with the following state-of-the-art algorithms: MajorVote [5], which trains the network with the majority voting labels from all the annotators. CrowdLayer [8], which directly learns from multiple annotators' labels with multiple annotator-specific linear layers. DoctorNet [1] which models multiple annotators individually with different softmax output layers in the deep architecture. In this baseline, all the annotators are equally weighted. WDN [1], which is a variant of DocterNet with learnable weights for different annotators. SpeeLFC [9], which is a novel end-to-end probabilistic model that learns interpretable parameters of the crowd layer.

We conduct all the experiments on NVIDIA RTX 2080Ti GPUs. Specifically, we repeat the experiments five times with different random seeds and calculate the average test accuracy and standard deviation for reporting the results. In future work, we would like to implement our UnionNet method using Mindspore [48], which is a new deep learning computing framework.

C. Training Settings

1) Network Structure: We use the same base model as the classifier for all algorithms. Following [8], we employ the four-layer CNN network as the classifier on Dogs versus Cats and LUNA16. We use VGG-16 as the classifier on CIFAR10 and use two-layer MLP on MNIST. On LabelMe, we also follow [8] to use the pre-trained VGG-16 model and apply an FC layer (with 128 units and ReLU activation) and an output layer on top with 50% dropout. On MGC, we use a linear model following the setting of SpeeLFC [9].

2) Optimizer: We use the Adam optimizer [49] for all the experiments. The hyper-parameters like learning rate for different datasets are provided in Table II.

D. Experimental Results

1) Synthesized Datasets: Table III shows the test accuracy of each algorithm on the four synthesized datasets under different crowdsourced settings. As we can see, our proposed algorithms achieve the best performance in most cases. On MNIST, UnionNet-B is the only algorithm that keeps the best or comparable performance across all the four cases, while CrowdLayer and UnionNet-A perform bad in the correlated

TABLE III

AVERAGE TEST ACCURACY (%) WITH STANDARD DEVIATION ON SYNTHESIZED DATASETS UNDER DIFFERENT SETTINGS (OVER FIVE TRIALS). THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Datasets	Settings	Baselines					Ours	
		MajorVote	CrowdLayer	DoctorNet	WDN	SpeeLFC	UnionNet-A	UnionNet-B
MNIST	Independent(H)	74.98±1.47	98.29±0.07	75.63±0.70	76.71±0.65	50.05±0.03	94.37±2.22	**98.39±0.05**
	Correlated(H)	83.43±1.28	78.58±11.15	74.92±0.75	56.10±0.80	50.03±0.04	79.57±1.83	**98.37±0.06**
	Independent(L)	89.94±1.01	95.58±0.15	93.97±0.25	93.77±0.97	95.51±0.90	95.54±0.16	**95.62±0.15**
	Correlated(L)	91.86±0.54	95.45±0.23	93.80±0.29	93.34±0.46	95.31±0.36	**95.55±0.22**	95.38±0.20
CIFAR10	Independent(H)	68.25±0.79	56.06±6.03	69.98±0.23	45.42±0.25	40.21±3.61	52.60±1.27	**78.40±3.35**
	Correlated(H)	67.81±2.67	68.31±5.62	67.77±1.90	45.32±0.11	35.39±0.17	45.10±0.62	**76.99±0.43**
	Independent(L)	60.95±0.82	68.21±1.11	66.07±1.41	46.52±4.97	66.37±1.46	67.49±0.42	**68.66±0.85**
	Correlated(L)	60.10±0.85	65.30±0.96	63.01±3.83	39.17±3.46	64.63±1.26	63.20±1.38	**65.82±1.06**
Dog vs. Cat	Independent(H)	77.62±0.37	**78.84±0.48**	76.87±0.2	77.13±0.75	78.61±0.62	78.60±0.51	78.66±0.83
	Correlated(H)	77.63±0.55	78.64±0.71	77.07±0.43	75.26±1.12	78.65±0.60	**78.79±0.66**	78.65±0.50
	Independent(L)	60.21±0.76	**70.05±0.58**	65.77±1.44	65.45±0.67	69.69±1.08	69.71±0.44	69.74±0.37
	Correlated(L)	61.25±1.11	68.44±1.17	65.49±1.07	64.73±0.95	68.58±1.23	68.54±0.56	**68.59±1.17**
LUNA16	Independent(H)	90.59±0.43	91.10±0.36	89.66±0.33	89.76±0.42	91.27±0.27	**91.29±0.39**	91.03±0.12
	Correlated(H)	90.45±0.22	91.00±0.18	89.84±0.30	89.65±0.33	91.05±0.40	90.95±0.26	**91.13±0.19**
	Independent(L)	88.05±0.89	89.10±0.55	87.90±0.73	88.01±0.71	**89.53±0.55**	89.44±0.41	89.40±0.39
	Correlated(L)	87.52±1.08	89.03±0.37	87.72±0.66	87.88±0.82	88.99±0.53	88.95±0.32	**89.17±0.51**

(H) case. Surprisingly, we observed an abnormal phenomenon that almost all algorithms except UnionNet-B provide worse results on two high-expertise cases than on the low-expertise cases. Recall that we assume that the annotators' expertise in situation (H) may vary across different classes. Consequently, the results we obtained seem to indicate that on multiclass datasets learning with noisy high-expertise annotations that needs to take into account class-level differences in expertise might be even more challenging than learning with simple low-expertise annotations (i.e., high noise rate).

On CIFAR10, UnionNet-B significantly outperforms all baselines in all the four settings, especially for experts with relatively high expertise. Specifically, UnionNet-B achieves significant improvement by about 12% over the second-best algorithms across the two cases with high expertise. In the two cases with low expertise, we observe that UnionNet-B still achieves the best performance while MajorVote performs much worse than the other baselines. It is worth noting that SpeeLFC performs badly on the two cases with relatively high expertise of both the MNIST dataset and CIFAR10 dataset, which shows the poor applicability of SpeeLFC to different cases.

On Dog versus Cat and LUNA16 datasets, we observe that all the methods exhibit similarly high accuracy in the two cases with high expertise. A possible reason is that there are only two classes for this dataset. An interesting phenomenon is that our proposed algorithms consistently outperform all the baselines under all the settings with correlated mistakes, while they are slightly inferior but still comparable to the best baseline in three of the settings with independent mistakes. It shows that our proposed method has an advantage in handling correlated mistakes, which are closer to real-world settings. This phenomenon can be interpreted by that our UnionNet can naturally coordinate the contributions of different annotators on the true label and explores the relationships between the annotators, while previous methods treat each annotator independently. Therefore, our UnionNet is better suited for correlated mistakes than independent mistakes. When it comes to the settings with independent mistakes, our UnionNet would achieve comparable performance to CrowdLayer since there is no relationship between the annotators.

TABLE IV

AVERAGE TEST ACCURACY (%) AND STANDARD DEVIATION (OVER FIVE TRIALS) ON REAL-WORLD DATASETS

Method	LabelMe	MGC
MajorVote	79.81±0.12	62.93±0.13
CrowdLayer	83.94±0.25	69.00±0.63
DoctorNet	79.97±0.40	57.07±0.25
WDN	81.08±0.22	41.47±0.50
SpeeLFC	83.70±0.33	48.40±3.22
UnionNet-A	83.92±0.21	68.60±0.53
UnionNet-B	**84.18±0.12**	**69.33±0.58**

2) Real-World Datasets: We also demonstrate the efficacy of the proposed methods on the real-world crowdsource datasets using LabelMe and MGC datasets in Table IV. On LabelMe Dataset, UnionNet-B gets the best result, while the performance of UnionNet-A is comparable to CrowdLayer and SpeeLFC. On MGC dataset, UnionNet-B still performs the best among all the method and UnionNet-A obtains comparable result with CrowdsLayer, while SpeeLFC is stuck at local minimum with poor performance.

To demonstrate the ability of our UnionNet to learn the reliabilities of the annotators, we compare the learned weight matrices of UnionNet-A and UnionNet-B with those of CrowdLayer, SpeeLFC, and the corresponding real confusion matrix on LabelMe dataset. Following [8], we select four annotators with the largest number of annotations. The results are shown in Fig. 2. The real transition matrices of annotators are calcu-

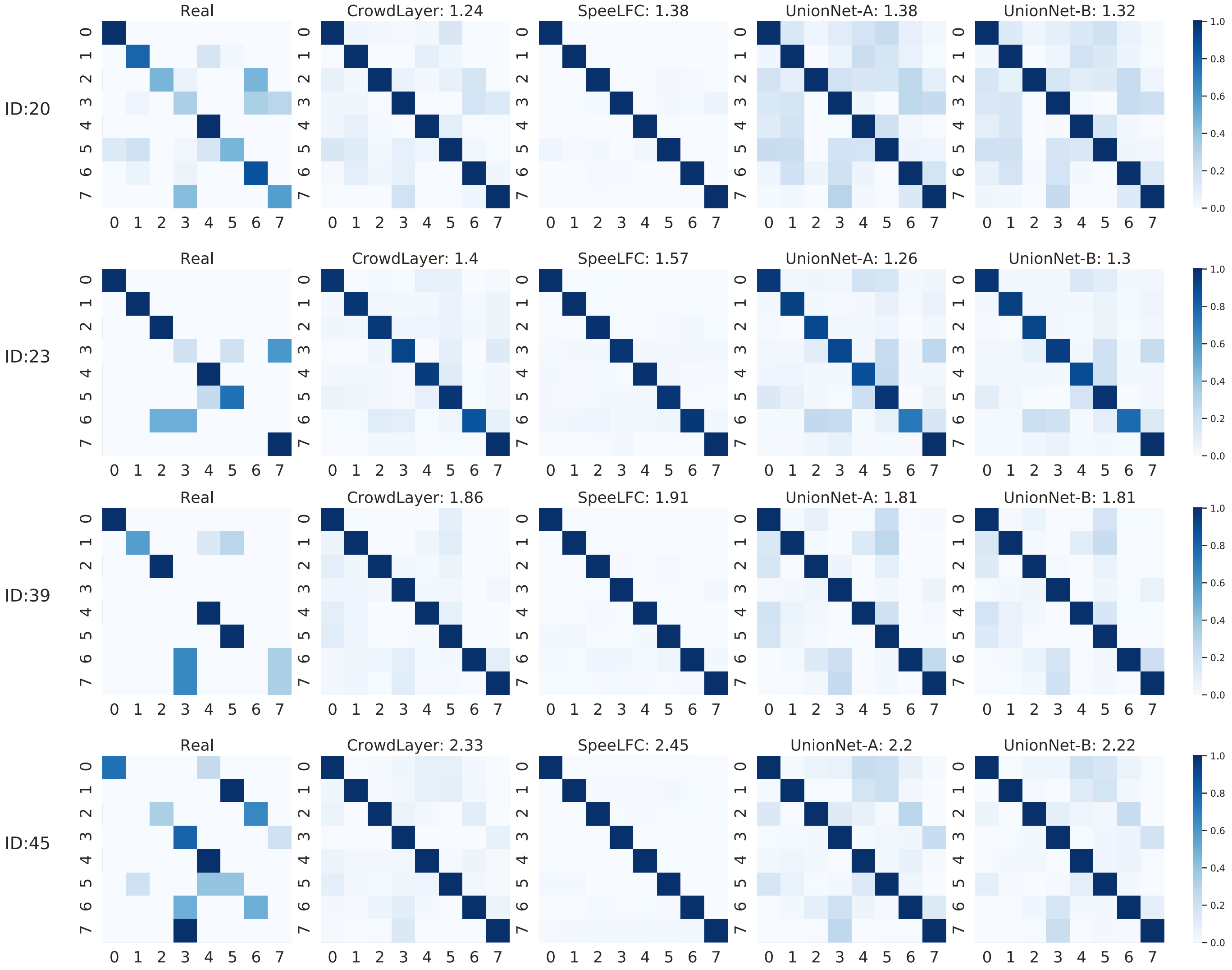


Fig. 2. Four examples in LabelMe for comparisons between the learned transition matrix of different algorithms and the real transition matrix. Note that the color intensity of the cells increases with the relative magnitude. The value behind the name of each algorithm is defined as $\|\mathbf{T}^{j}_{\mathrm{alg}} - \mathbf{T}^{j}_{\mathrm{real}}\|_{F}$, where $\mathbf{T}^{j}_{\mathrm{alg}}$ and $\mathbf{T}^{j}_{\mathrm{real}}$ denote the learned weight matrix and the corresponding real confusion matrices for the jth annotator. The smaller the value, the better the fitting performance.

lated based on their annotations and the ground truth, while the learned weight matrices of every algorithm are normalized for fair comparisons. From Fig. 2, we observe that the learned matrices of UnionNet-A and UnionNet-B are closer to the real confusion matrix than those of CrowdLayer and SpeeLFC. This observation validates the effectiveness of our UnionNet, as UnionNet could automatically coordinate the influences of different annotators on the true label, while other algorithms only focus on each annotator independently without explicit consideration of the relationships among the annotators.

E. Ablation Study

To demonstrate the advantage of our concatenation operation over the addition operation, we set up the experiments above LabelMe and MGC datasets. For reimplementing UnionNet-A and UnionNet-B with an addition operation, we initialize the square transition matrix by (7) on LabelMe. Slightly different from (8), the transition matrix on MGC is initialized by taking logarithm before addition. The results are presented in Fig. 3. On both LabelMe and MGC datasets, UnionNet-A and UnionNet-B with concatenation operation consistently perform better than the corresponding algorithms based on the addition operation, respectively. It verifies that, compared to the addition operation, the concatenation operation is more effective to build the union of crowdsourced labels for training.

F. Training Efficiency Analysis

To demonstrate the advantage of UnionNet in training efficiency, we train each algorithm for ten epochs on real-world datasets and show the average training time per epoch in Fig. 4. Besides, we introduce a new baseline named Standard, which simply trains on the ground truth label. We omit WDN in the comparisons as it is one of the variants of DoctorNet and its training time is much higher than that of the DoctorNet.

On LabelMe dataset, CrowdLayer, DoctorNet, and SpeeLFC take eight and nine times longer than Standard and MajorVote. In contrast, the training speed of both UnionNet-A and

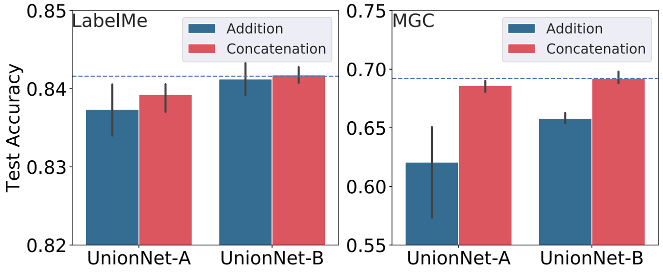


Fig. 3. Average test accuracy (%) with standard deviation on LabelMe (left) and MGC (right) over five trials, for different implementations of UnionNet.

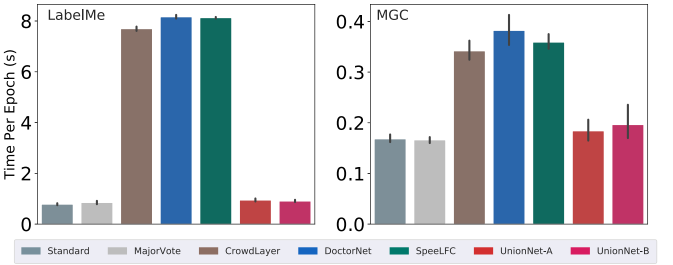


Fig. 4. Average training time per epoch (over ten epochs) on LabelMe (left) and MGC (right) for different algorithms.

UnionNet-B is close to that of Standard and MajorVote, which means the linear layer proposed in our algorithms runs with almost no additional computational burden.

The same phenomenon happens on MGC dataset. Due to their for-loop forward propagation in multiple annotator-specific transition matrices, DoctorNet spends the most time in training, followed by CrowdLayer and SpeeLFC. On the other hand, learning directly from the concatenated labels through a simple linear layer costs very little on computational resources, shown by UnionNet-A and UnionNet-B. It shows that our proposed UnionNet is an end-to-end learning architecture with higher computational efficiency compared with other existing end-to-end learning algorithms.

VII. Conclusion

In this article, we investigated the problem of learning with crowdsourced labels and proposed a novel end-to-end deep learning method named UnionNet, which is not only theoretically consistent but also experimentally effective and efficient. Specifically, unlike existing methods that either fit the given label by each annotator independently or fuse all the labels into a reliable one, we concatenated the one-hot encoded vectors of crowdsourced labels provided by all the annotators, which takes all the labeling information as a union and keeps it intact and coordinates multiple annotators. In this way, we could directly train an end-to-end deep neural network by maximizing the likelihood of this union with only a parametric transition matrix. We theoretically proved the learning consistency and experimentally showed the effectiveness and the efficiency of our proposed method. In future work, we will extend UnionNet to the regression and structured prediction problems.

References

[1] M. Y. Guan, V. Gulshan, A. M. Dai, and G. E. Hinton, "Who said what: Modeling individual labelers improves classification," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3109–3118.

[2] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, and N. Navab, "AggNet: Deep learning from crowds for mitosis detection in breast cancer histology images," *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1313–1321, May 2016.

[3] J. Li, C. Zhang, J. T. Zhou, H. Fu, S. Xia, and Q. Hu, "Deep-LIFT: Deep label-specific feature learning for image annotation," *IEEE Trans. Cybern.*, early access, Feb. 10, 2021, doi: 10.1109/TCYB.2021.3049630.

[4] F. Rodrigues, F. Pereira, and B. Ribeiro, "Learning from multiple annotators: Distinguishing good from random labelers," *Pattern Recognit. Lett.*, vol. 34, no. 12, pp. 1428–1436, Sep. 2013.

[5] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.

[6] B. Han, I. W. Tsang, L. Chen, J. T. Zhou, and C. P. Yu, "Beyond majority voting: A coarse-to-fine label filtration for heavily noisy labels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 12, pp. 3774–3787, Dec. 2019.

[7] V. C. Raykar *et al.*, "Learning from crowds," *J. Mach. Learn. Res.*, vol. 11, no. 4, pp. 1297–1322, 2010.

[8] F. Rodrigues and F. C. Pereira, "Deep learning from crowds," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1611–1618.

[9] Z. Chen *et al.*, "Structured probabilistic end-to-end learning from crowds," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, Jul. 2020, pp. 1512–1518.

[10] J. Whitehill, T.-F. Wu, J. Bergsma, J. R. Movellan, and P. L. Ruvolo, "Whose vote should count more: Optimal integration of labels from labelers of unknown expertise," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2035–2043.

[11] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *J. Roy. Stat. Soc., C (Appl. Statist.)*, vol. 28, no. 1, pp. 20–28, 1979.

[12] F. Rodrigues, F. Pereira, and B. Ribeiro, "Gaussian process classification and active learning with multiple annotators," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 433–441.

[13] F. Rodrigues, M. Lourenço, B. Ribeiro, and F. C. Pereira, "Learning supervised topic models for classification and regression from crowds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2409–2422, Dec. 2017.

[14] P. Cao, Y. Xu, Y. Kong, and Y. Wang, "Max-MIG: An information theoretic approach for joint learning from crowds," 2019, *arXiv:1905.13436*.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[16] Y. Yan, R. Rosales, G. Fung, R. Subramanian, and J. Dy, "Learning from multiple annotators with varying expertise," *Mach. Learn.*, vol. 95, no. 3, pp. 291–327, 2014.

[17] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, Jul. 2007.

[18] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Jun. 2014.

[19] H. Yang, J. T. Zhou, Y. Zhang, B.-B. Gao, J. Wu, and J. Cai, "Exploit bounding box annotations for multi-label object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 280–288.

[20] H. Yang, J. T. Zhou, J. Cai, and Y. S. Ong, "MIML-FCN+: Multi-instance multi-label learning via fully convolutional networks with privileged information," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1577–1585.

[21] H. Yang, J. T. Zhou, and J. Cai, "Improving multi-label learning with missing labels by structured semantic correlations," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 835–851.

[22] E. L. Lehmann and G. Casella, *Theory of Point Estimation*. Springer, 2006.

[23] Y. Zhang, N. Charoenphakdee, and M. Sugiyama, "Learning from indirect observations," 2019, *arXiv:1910.04394*.

[24] 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.

[25] B. Han *et al.*, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8527–8537.

[26] X. Ma *et al.*, "Dimensionality-driven learning with noisy labels," 2018, *arXiv:1806.02612*.

[27] B. Han *et al.*, "Masking: A new perspective of noisy supervision," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5836–5846.

[28] X. Xia *et al.*, "Are anchor points really indispensable in label-noise learning?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 6838–6849.

[29] 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.

[30] H. Wei, L. Tao, R. Xie, and B. An, "Open-set label noise can improve robustness against inherent label noise," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 7978–7992.

[31] Y. Yao *et al.*, "Jo-SRC: A contrastive approach for combating noisy labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5192–5201.

[32] Z. Sun, X.-S. Hua, Y. Yao, X.-S. Wei, G. Hu, and J. Zhang, "CRSSC: Salvage reusable samples from noisy data for robust learning," in *Proc. 28th ACM Int. Conf. Multimedia*, Oct. 2020, pp. 92–101.

[33] Z. Sun *et al.*, "Webly supervised fine-grained recognition: Benchmark datasets and an approach," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10602–10611.

[34] J. Krause *et al.*, "The unreasonable effectiveness of noisy data for fine-grained recognition," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 301–320.

[35] 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.

[36] P. Hu, X. Peng, H. Zhu, L. Zhen, and J. Lin, "Learning cross-modal retrieval with noisy labels," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 5403–5413.

[37] M. Yang, Y. Li, Z. Huang, Z. Liu, P. Hu, and X. Peng, "Partially view-aligned representation learning with noise-robust contrastive loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 1134–1143.

[38] X. Yu, T. Liu, M. Gong, and D. Tao, "Learning with biased complementary labels," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 68–83.

[39] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," in *Proc. Int. Conf. Learn. Represent.*, 2017.

[40] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. Cambridge, MA, USA: MIT Press, 2012.

[41] A. Maurer, "A vector-contraction inequality for Rademacher complexities," in *Proc. Int. Conf. Algorithmic Learn. Theory*. Springer, 2016, pp. 3–17.

[42] N. Lu, T. Zhang, G. Niu, and M. Sugiyama, "Mitigating overfitting in supervised classification from two unlabeled datasets: A consistent risk correction approach," in *Proc. AISTATS*, 2020, pp. 1115–1125.

[43] N. Golowich, A. Rakhlin, and O. Shamir, "Size-independent sample complexity of neural networks," 2017, *arXiv:1712.06541*.

[44] S. Mendelson, "Lower bounds for the empirical minimization algorithm," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3797–3803, Aug. 2008.

[45] K. Atarashi, S. Oyama, and M. Kurihara, "Semi-supervised learning from crowds using deep generative models," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1555–1562.

[46] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality data?" Tech. Rep., 2016.

[47] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "LabelMe: A database and web-based tool for image annotation," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 157–173, 2008.

[48] Huawei. (2020). *Mindspore*. [Online]. Available: https://www.mindspore.cn/

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

Hongxin Wei received the B.E. degree in software engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2016. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

His research interests include weakly supervised learning, adversarial robustness, OOD detection, and data mining.

Mr. Wei has served as a Program Committee Member (or a Reviewer) for NeurIPS, ICLR, KDD, CVPR, and ICML.

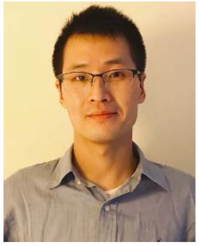
Renchunzi Xie received the B.Ec. degree in economic statistics from the Southwestern University of Finance and Economics, Chengdu, China, in 2017, and the M.S. degree in data science from University College London, London, U.K., in 2018. She is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

Her research interests include partial-label learning and domain adaptation.

Lei Feng received the Ph.D. degree in computer science from Nanyang Technological University, Singapore, in 2021.

He is currently a Professor with the College of Computer Science, Chongqing University, Chongqing, China. He was named to Forbes 30 under 30 China 2021. He has published over 20 papers on top conferences, such as ICML, NeurIPS, ICLR, KDD, CVPR, ICCV, AAAI, and IJCAI. His research interests include weakly supervised learning, statistical learning theory, and data mining.

Dr. Feng has served as a Senior Program Committee Member for IJCAI 2021 and AAAI 2022, an Expert Review for ICML 2021, and a program committee member (or a reviewer) for other top conferences and journals.



Bo Han received the Ph.D. degree in computer science from the University of Technology at Sydney, Sydney, NSW, Australia, in 2019.

From 2018 to 2019, he was a Research Intern with the AI Residency Program, RIKEN Center for Advanced Intelligence Project (RIKEN AIP), Tokyo, Japan. He was a Post-Doctoral Fellow with RIKEN AIP from 2019 to 2020. He is currently an Assistant Professor of computer science with Hong Kong Baptist University, Hong Kong, and a Visiting Scientist with RIKEN AIP. His research interests include machine learning and deep learning.

Dr. Han received the RIKEN BAIHO Award in 2019, the RGC Early Career Scheme in 2020, and the MSRA StarTrack Program in 2021. He has served as the Area Chair of NeurIPS, ICML, and ICLR, and the Action Editor of *Transactions on Machine Learning Research* and *Neural Networks*.

Bo An received the Ph.D. degree in computer science from the University of Massachusetts at Amherst, Amherst, MA, USA, in 2010.

He is currently the President's Council Chair Associate Professor of computer science and engineering and the Co-Director of the Artificial Intelligence Research Institute (AI.R), Nanyang Technological University, Singapore. He has published over 100 referred papers at AAMAS, IJCAI, AAAI, ICAPS, KDD, UAI, EC, WWW, ICLR, NeurIPS, ICML, JAAMAS, AIJ, and ACM/IEEE TRANSACTIONS. His current research interests include artificial intelligence, multiagent systems, computational game theory, reinforcement learning, and optimization.

Dr. An was a recipient of the 2010 IFAAMAS Victor Lesser Distinguished Dissertation Award, the Operational Excellence Award from the Commander, First Coast Guard District of the United States, the 2012 INFORMS Daniel H. Wagner Prize for Excellence in Operations Research Practice, and the 2018 Nanyang Research Award (Young Investigator). His publications won the Best Innovative Application Paper Award at AAMAS'12 and the Innovative Application Award at IAAI'16. He was invited to give Early Career Spotlight talk at IJCAI'17. He led the team HogRider which won the 2017 Microsoft Collaborative AI Challenge. He was named to IEEE Intelligent Systems' "AI's 10 to Watch" list for 2018. He is also the PC Co-Chair of AAMAS'20. He is also a member of the Editorial Board of *Journal of Artificial Intelligence Research* (JAIR) and the Associate Editor of JAAMAS, *IEEE Intelligent Systems*, and *ACM Transactions on Intelligent Systems and Technology* (TIST). He was elected to the Board of Director of IFAAMAS, a Senior Member of AAAI, and a Distinguished Member of ACM.