ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240829

Annot-Mix: Learning with Noisy Class Labels from Multiple Annotators via a Mixup Extension

Marek Herde[a,*], Lukas Lührs[a], Denis Huseljic[a] and Bernhard Sick[a]

[a]University of Kassel, Germany

Abstract. Training with noisy class labels impairs neural networks' generalization performance. In this context, `mixup` is a popular regularization technique to improve training robustness by making memorizing false class labels more difficult. However, `mixup` neglects that multiple annotators, e.g., crowdworkers, typically provide class labels. Therefore, we propose an extension of `mixup`, which handles multiple class labels per instance while considering which class label originates from which annotator. Integrated into our multi-annotator classification framework `annot-mix`, it performs superiorly to eleven (mostly state-of-the-art) approaches in an evaluation study with eleven datasets comprising noisy class labels from either human or simulated annotators. Our code is publicly available through our GitHub repository at https://github.com/ies-research/multi-annotator-machine-learning/tree/annot-mix.

1 Introduction

Training machine learning models, such as deep neural networks (DNNs), to solve classification tasks requires data instances with associated class labels, typically acquired from human annotators, e.g., crowdworkers [43], in a labor-intensive process. Such annotators may be prone to errors for various reasons, e.g., lack of domain expertise, exhaustion, or disinterest [18]. The resulting annotation errors, called noisy class labels [12], impair NNs' generalization performance because NNs easily overfit on training data by memorizing noisy class labels [37]. Consequently, various approaches have been proposed to address this issue. A popular data augmentation and regularization technique is `mixup` [50], whose idea is to generate convex combinations of pairs of instances and their respective class labels (cf. the first and second column of Fig. 1 for an example in a standard classification setting). This widely applicable augmentation during the training of NNs makes pure memorization more difficult and thus reduces sensitivity to class label noise. Despite its simplicity and effectiveness, `mixup` has not been fully extended to classification tasks with class labels provided by multiple annotators, often referred to as multi-annotator classification [19] or learning from crowds [33]. In this context, we face two major challenges:

- `mixup` ignores that multiple class labels from varying numbers of annotators can be assigned to a single data instance.
- `mixup` ignores which class label originates from which annotator.

Motivated by these challenges, Fig. 1 formulates our central research question, which we address through the following contributions:

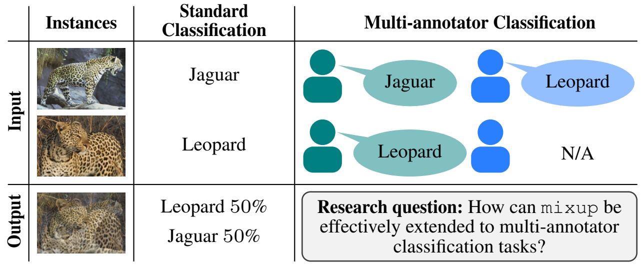

Figure 1. Illustration of vanilla `mixup` and our research question: In the standard classification tasks (cf. first and second column), `mixup` convexly combines the two animal images (cf. acknowledgments for crediting USFWS) and their class labels. In contrast, multi-annotator classification tasks (cf. first and third column) allow multiple class labels to be assigned to a single instance. Further, we know which class label originates from which annotator, and some class labels may not be available (N/A) from some annotators. Hence, we must extend `mixup` toward such tasks.

- We propose a `mixup` extension that handles multiple class labels per instance and considers each label's annotator.
- We integrate this extension into our multi-annotator classification approach `annot-mix`, which estimates each annotator's performance while training an NN as a classification model.
- We present an extensive experimental evaluation study demonstrating the superior performance of `annot-mix` compared to eleven mostly state-of-the-art approaches across image, text, and tabular datasets with either human or simulated noisy class labels.

This article's remainder is structured as follows: Section 2 formally introduces the problem setup of multi-annotator classification tasks. Subsequently, we discuss related work of multi-annotator classification and existing variants of `mixup` in Section 3. Section 4 presents our approach `annot-mix`, which is evaluated, including an ablation and hyperparameter study, in Section 5. We conclude this work with an outlook on future research in Section 6.

2 Problem Setup

Figure 2 depicts the probabilistic graphical model that overviews the random variables and their dependencies of the commonly assumed data generation process in multi-annotator classification [19, 26]. More concretely, there is a multi-set $\mathcal{X} := \{\boldsymbol{x}_n\}_{n=1}^{N} \subset \Omega_X := \mathbb{R}^D$ of $N \in \mathbb{N}_{>0}$ instances as $D \in \mathbb{N}_{>0}$-dimensional vectors, which are independently sampled from the categorical distribution $\Pr(\boldsymbol{x})$. Their $C \in \mathbb{N}_{>1}$-dimensional one-hot encoded true class labels form

* Corresponding Author. Email: marek.herde@uni-kassel.de.

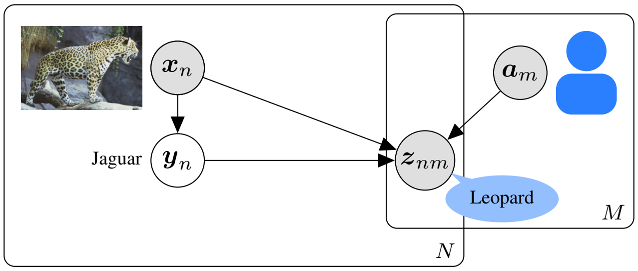

Figure 2. Probabilistic graphical model of the data generation in multi-annotator classification: Arrows indicate dependencies between random variables, shaded circles observable random variables, and white circles latent random variables.

a multi-set $\mathcal{Y} := \{\boldsymbol{y}_n\}_{n=1}^N \subseteq \Omega_Y := \{\boldsymbol{e}_c\}_{c=1}^C$, where C denotes the number of classes. In standard classification, a class label $\boldsymbol{y}_n$ is observed and sampled from the distribution $\Pr(\boldsymbol{y} \,|\, \boldsymbol{x}_n)$. However, in multi-annotator classification, we do not know the true class labels $\mathcal{Y}$. Instead, $M \in \mathbb{N}_{>1}$ error-prone annotators, denoted as a multi-set of vectors $\mathcal{A} := \{\boldsymbol{a}_m\}_{m=1}^M \subset \Omega_A$, provide independently from each other noisy class labels. Throughout this article, we identify each annotator $\boldsymbol{a}_m$ by an M-dimensional one-hot encoded vector $\boldsymbol{e}'_m$ such that $\Omega_A := \{\boldsymbol{e}'_1, \ldots, \boldsymbol{e}'_M\}$. In principle, other representations would also be conceivable if metadata about the annotators [52] were available, e.g., annotators' levels of education. The noisy class labels are denoted as the multi-set $\mathcal{Z} := \{\boldsymbol{z}_{nm}\}_{n=1,m=1}^{N,M} \subseteq \Omega_Z := \Omega_Y \cup \{\boldsymbol{0}\}$. Thereby, $\boldsymbol{z}_{nm} \in \Omega_Y$, sampled from the distribution $\Pr(\boldsymbol{z} \,|\, \boldsymbol{x}_n, \boldsymbol{y}_n, \boldsymbol{a}_m)$, is the class label assigned by annotator $\boldsymbol{a}_m$ to instance $\boldsymbol{x}_n$ with the true class label $\boldsymbol{y}_n$. In the case of $\boldsymbol{z}_{nm} = \boldsymbol{0}$, the annotator $\boldsymbol{a}_m$ has not annotated instance $\boldsymbol{x}_n$, e.g., due to a limited annotation budget [24].

Based on the above setup, the objective in multi-annotator classification tasks is as follows:

Objective: Given instances $\mathcal{X}$, annotators $\mathcal{A}$, and noisy class labels $\mathcal{Z}$, we aim to train a classification model $\boldsymbol{y}_{\boldsymbol{\theta}^\star} : \Omega_X \to \Omega_Y$ with parameters $\boldsymbol{\theta}^\star \in \Theta$, which maximizes the accuracy:

$$\boldsymbol{\theta}^\star := \underset{\boldsymbol{\theta} \in \Theta}{\arg\max} \left(\mathrm{E}_{\boldsymbol{x},\boldsymbol{y}}[\boldsymbol{y}^\mathrm{T} \boldsymbol{y}_{\boldsymbol{\theta}}(\boldsymbol{x})] \right). \tag{1}$$

3 Related Work

Learning from noisy class labels is a highly relevant research area [1, 12, 37]. Here, we focus on one- and two-stage approaches [26] in the multi-annotator classification setup (cf. Section 2) and robust regularization approaches [37], to which `mixup` belongs.

3.1 Multi-annotator Classification

Two-stage multi-annotator classification approaches approximate true class labels by aggregating multiple noisy class labels per instance. Subsequently, the aggregated class labels and their associated instances serve as the training dataset for the downstream task. The simplest aggregation approach is majority voting, which outputs the class label with the most annotator votes per instance. By doing so, majority voting naively assumes all annotators have the same accuracy [5, 23]. More advanced approaches [5, 8, 23, 41] overcome this issue by estimating each annotator's performance when aggregating class labels. For example, the Dawid-Skene algorithm [8] estimates a confusion matrix per annotator. However, such approaches typically expect multiple class labels for each instance [24].

One-stage multi-annotator classification approaches do not need multiple class labels per instance because they train the classification model without any detached stage for aggregating class labels. A common training principle is to leverage the expectation-maximization (EM) algorithm that iteratively updates the classification model's parameters and annotators' performance estimates (M-step) to accurately estimate the latent true class labels (E-step) [24, 33, 47]. Such EM algorithms come at the price of high computational complexity and the need to decide when to switch between E- and M-steps [34]. Therefore, several approaches have been proposed to overcome these issues when training NNs. A common approach is to extend an NN-based classification model by a noise adaption layer [34, 45], whose parameters encode annotators' performances on top of the classification layer. Alternatively, a separate so-called annotator (performance) model is jointly trained with the classification model [4, 6, 19, 40, 22]. In this case, both models' outputs are combined when optimizing the target loss. Alongside the training algorithm, the underlying assumptions regarding the modeling of annotator performance play a crucial role. Here, simplifications of the probabilistic graphical model in Fig. 2 are often made, for example, by ignoring the instance dependency of the annotator performance [34, 45, 40]. Our work follows recent one-stage approaches [4, 6, 19], which model annotator performance as a function of the latent true class label and the instance's features.

3.2 Robust Regularization

Regularization reduces NNs' overfitting on instances with false class labels. However, common regularization approaches, such as weight decay [25] and dropout [38], are often insufficient for tasks with severe class label noise [37]. Data augmentation via `mixup` is a more robust regularization approach [50]. Given two randomly drawn instances $\boldsymbol{x}_n, \boldsymbol{x}_{\hat{n}} \in \mathcal{X}$ with their true class labels $\boldsymbol{y}_n, \boldsymbol{y}_{\hat{n}} \in \mathcal{Y}$, a new instance-label pair for training is generated via convex combination:

$$\widetilde{\boldsymbol{x}} := \lambda \boldsymbol{x}_n + (1-\lambda)\boldsymbol{x}_{\hat{n}}, \quad \widetilde{\boldsymbol{y}} := \lambda \boldsymbol{y}_n + (1-\lambda)\boldsymbol{y}_{\hat{n}}, \tag{2}$$

where the scalar $\lambda \in [0,1]$ is sampled from a symmetrical beta distribution $\mathrm{Beta}(\alpha, \alpha)$ with the concentration parameter $\alpha \in \mathbb{R}_{>0}$. This way, `mixup` expands the training dataset by utilizing the idea that interpolating feature vectors linearly should result in linear interpolations of their targets, requiring minimal implementation and computational overhead. Meanwhile, various extensions of `mixup` have been proposed, including an extension mixing hidden states of NNs [44] and an extension specifically tailored for image [49] or text data [39]. To the best of our knowledge, the only `mixup` extension [54] for multiple annotators is proposed for opinion expression identification tasks [3]. Its idea is to make predictions by combining learned annotator embeddings for the same instance to simulate the annotation of an expert. Beyond the task type, our approach differs substantially from this extension by mixing class labels across different instances and annotators while explicitly modeling each annotator's performance.

4 The Annot-Mix Approach

This section presents our one-stage multi-annotator classification approach `annot-mix`, which we train through marginal likelihood maximization while leveraging our novel `mixup` extension for robust regularization. Figure 3 overviews our approach's architecture.

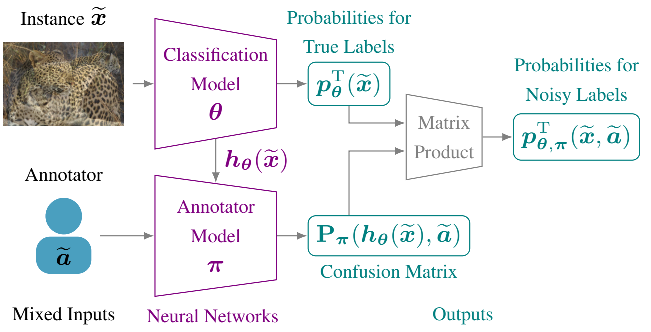

Figure 3. Architecture and forward propagation of `annot-mix`: The inputs, obtained after using our `mixup` extension, are propagated through the classification and annotator model, whose outputs are combined to obtain the probabilities of the observed noisy class labels.

4.1 *Marginal Likelihood Maximization*

Assuming the probabilistic graphical model of Fig. 2, the joint distribution of the true class label $\boldsymbol{y}$ and noisy class label $\boldsymbol{z}$ given the instance $\boldsymbol{x}_n$ and the annotator $\boldsymbol{a}_m$ factors into a product of two categorical distributions:

$$\Pr(\boldsymbol{y},\boldsymbol{z}\,|\,\boldsymbol{x}_n,\boldsymbol{a}_m) = \Pr(\boldsymbol{y}\,|\,\boldsymbol{x}_n)\cdot\Pr(\boldsymbol{z}\,|\,\boldsymbol{x}_n,\boldsymbol{a}_m,\boldsymbol{y}). \quad (3)$$

For predicting instances' class labels, we aim to estimate an instance's class-membership probability distribution $\Pr(\boldsymbol{y}\,|\,\boldsymbol{x}_n)$. Therefor, we employ a classification model in the form of an NN with parameters $\boldsymbol{\theta}\in\Theta$, defined through the function

$$\boldsymbol{p}_{\boldsymbol{\theta}}:\Omega_X\rightarrow\Delta_C\coloneqq\{\boldsymbol{p}\in[0,1]^C\,|\,||\boldsymbol{p}||_1=1\}, \quad (4)$$

where $||\cdot||_1$ denotes the 1-norm and $\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)$ are the estimated probabilities for the true class label of instance $\boldsymbol{x}_n$. Accordingly, the estimated Bayes optimal prediction for our objective in Eq. (1) is given by the class with the maximum probability estimate:

$$\boldsymbol{y}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)\coloneqq\underset{\boldsymbol{e}_c\in\Omega_Y}{\arg\max}\left(\boldsymbol{e}_c^{\mathrm{T}}\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)\right). \quad (5)$$

For estimating annotators' performances, we aim to approximate the probability distribution $\Pr(\boldsymbol{z}\,|\,\boldsymbol{x}_n,\boldsymbol{a}_m,\boldsymbol{y})$ for each possible class label $\boldsymbol{y}\in\Omega_Y$. Therefor, we employ an annotator model in the form of an NN with parameters $\boldsymbol{\pi}\in\Pi$, defined through the function

$$\mathbf{P}_{\boldsymbol{\pi}}:\Omega_H\times\Omega_A\rightarrow\{(\boldsymbol{p}_1,\ldots,\boldsymbol{p}_C)^{\mathrm{T}}\,|\,\boldsymbol{p}_1,\ldots,\boldsymbol{p}_C\in\Delta_C\}, \quad (6)$$

where $\mathbf{P}_{\boldsymbol{\pi}}(\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_n),\boldsymbol{a}_m)$ is the estimated confusion matrix of annotator $\boldsymbol{a}_m$ for instance $\boldsymbol{x}_n$, represented through $\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)\in\Omega_H$ as output of the classification model's penultimate layer.

Since the true class labels $\mathcal{Y}$ in the complete likelihood function $\Pr(\mathcal{Y},\mathcal{Z}\,|\,\mathcal{X},\mathcal{A};\boldsymbol{\theta},\boldsymbol{\pi})$ are latent, we optimize both models' parameters $(\boldsymbol{\theta},\boldsymbol{\pi})$ by maximizing the marginal likelihood [19] of the observed noisy class labels $\mathcal{Z}'\coloneqq\{\boldsymbol{z}_{nm}\in\mathcal{Z}\,|\,\boldsymbol{z}_{nm}\neq\mathbf{0}\}$:

$$\Pr(\mathcal{Z}'\,|\,\mathcal{X},\mathcal{A};\boldsymbol{\theta},\boldsymbol{\pi}) = \prod_{\boldsymbol{x}_n\in\mathcal{X}}\prod_{\boldsymbol{a}_m\in\mathcal{A}_n}\Pr(\boldsymbol{z}_{nm}\,|\,\boldsymbol{x}_n,\boldsymbol{a}_m;\boldsymbol{\theta},\boldsymbol{\pi}) \quad (7)$$

$$= \prod_{\boldsymbol{x}_n\in\mathcal{X}}\prod_{\boldsymbol{a}_m\in\mathcal{A}_n}\left(\sum_{c=1}^{C}\begin{matrix}\Pr(\boldsymbol{y}=\boldsymbol{e}_c\,|\,\boldsymbol{x}_n;\boldsymbol{\theta})\,\cdot\\ \Pr(\boldsymbol{z}_{nm}\,|\,\boldsymbol{x}_n,\boldsymbol{a}_m,\boldsymbol{y}=\boldsymbol{e}_c;\boldsymbol{\pi})\end{matrix}\right) \quad (8)$$

$$= \prod_{\boldsymbol{x}_n\in\mathcal{X}}\prod_{\boldsymbol{a}_m\in\mathcal{A}_n}\underbrace{\boldsymbol{p}_{\boldsymbol{\theta}}^{\mathrm{T}}(\boldsymbol{x}_n)\mathbf{P}_{\boldsymbol{\pi}}(\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_n),\boldsymbol{a}_m)}_{\boldsymbol{p}_{\boldsymbol{\theta},\boldsymbol{\pi}}^{\mathrm{T}}(\boldsymbol{x}_n,\boldsymbol{a}_m)\coloneqq}\boldsymbol{z}_{nm}, \quad (9)$$

where $\mathcal{A}_n\coloneqq\{\boldsymbol{a}_m\in\mathcal{A}\,|\,\boldsymbol{z}_{nm}\neq\mathbf{0}\}$ comprises the annotators who provided a class label for instance $\boldsymbol{x}_n$. The marginalization (summation) of the latent true class label is shown in Eq. (8). The function $\boldsymbol{p}_{\boldsymbol{\theta},\boldsymbol{\pi}}:\Omega_X\times\Omega_A\rightarrow\Delta_C$ in Eq. (9) outputs probabilities estimating which class label an annotator will assign to an instance. Thus, the function $\boldsymbol{z}_{\boldsymbol{\theta},\boldsymbol{\pi}}:\Omega_X\times\Omega_A\rightarrow\Omega_Y$ outputting the class label with the highest estimated probability is given by:

$$\boldsymbol{z}_{\boldsymbol{\theta},\boldsymbol{\pi}}(\boldsymbol{x}_n,\boldsymbol{a}_m)\coloneqq\underset{\boldsymbol{e}_c\in\Omega_Y}{\arg\max}\left(\boldsymbol{e}_c^{\mathrm{T}}\boldsymbol{p}_{\boldsymbol{\theta},\boldsymbol{\pi}}(\boldsymbol{x}_n,\boldsymbol{a}_m)\right). \quad (10)$$

Converting the marginal likelihood function in Eq. (9) into a negative log-likelihood function yields the cross-entropy as the loss function:

$$L_{\mathcal{X},\mathcal{A},\mathcal{Z}'}(\boldsymbol{\theta},\boldsymbol{\pi})\coloneqq-\sum_{\boldsymbol{x}_n\in\mathcal{X}}\sum_{\boldsymbol{a}_m\in\mathcal{A}_n}\frac{\boldsymbol{z}_{nm}^{\mathrm{T}}\ln\left(\boldsymbol{p}_{\boldsymbol{\theta},\boldsymbol{\pi}}(\boldsymbol{x}_n,\boldsymbol{a}_m)\right)}{|\mathcal{Z}'|}, \quad (11)$$

where $\ln(\cdot)$ is applied elementwise to the input vector. By default, optimal predictions for $\boldsymbol{p}_{\boldsymbol{\theta}}(\boldsymbol{x}_n)$ and $\mathbf{P}_{\boldsymbol{\pi}}(\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_n),\boldsymbol{a}_m)$ are not identifiable [22] because there are multiple combinations to produce the same output $\boldsymbol{p}_{\boldsymbol{\theta},\boldsymbol{\pi}}(\boldsymbol{x}_n,\boldsymbol{a}_m)$. Therefore, we resort to a common solution proposed in literature [45, 19] and initialize the annotator model's parameters $\boldsymbol{\pi}$ to approximately satisfy:

$$\forall\boldsymbol{h}\in\Omega_H,\forall\boldsymbol{a}\in\Omega_A:\mathbf{P}_{\boldsymbol{\pi}}(\boldsymbol{h},\boldsymbol{a})\approx\eta\boldsymbol{I}_C+\frac{(1-\eta)}{C-1}\left(\mathbf{1}_C-\boldsymbol{I}_C\right), \quad (12)$$

with $\eta\in(0,1)$ as the probability of obtaining a correct class label, $\boldsymbol{I}_C\in\mathbb{R}^{C\times C}$ as an identity matrix, and $\mathbf{1}_C\in\mathbb{R}^{C\times C}$ as an all-one matrix. We set $\eta\coloneqq 0.9>{}^1/_C$, implying that solutions with diagonally dominant confusion matrices are preferred at training start.

4.2 *Mixup Extension*

Optimizing the loss function in Eq. (11) corresponds to empirical risk minimization (ERM) [42] because the classification and annotator models are forced to fit the observed noisy class labels perfectly. Although the annotator model attempts to separate the noise in the class labels during training, overfitting is still an issue (cf. Section 5). Therefore, we extend `mixup` [39] for robust regularization and improved generalization of `annot-mix`. Our idea for the extension of `mixup` to multi-annotator classification tasks lies in shifting the perspective from mixing tuples of instances and class labels to mixing triples of instances, annotators, and class labels. Concretely, we propose the following extension:

> Given two triples $(\boldsymbol{x}_n,\boldsymbol{a}_m,\boldsymbol{z}_{nm}),(\boldsymbol{x}_{\hat{n}},\boldsymbol{a}_{\hat{m}},\boldsymbol{z}_{\hat{n}\hat{m}})$, randomly sampled from
>
> $$\mathcal{M}\coloneqq\{(\boldsymbol{x}_n,\boldsymbol{a}_m,\boldsymbol{z}_{nm})\,|\,\boldsymbol{x}_n\in\mathcal{X},\boldsymbol{a}_m\in\mathcal{A},\boldsymbol{z}_{nm}\in\mathcal{Z}'\}, \quad (13)$$
>
> we `mixup` instances, annotators, and noisy class labels via
>
> $$\widetilde{\boldsymbol{x}}\coloneqq\lambda\boldsymbol{x}_n+(1-\lambda)\boldsymbol{x}_{\hat{n}}, \quad (14)$$
> $$\widetilde{\boldsymbol{a}}\coloneqq\lambda\boldsymbol{a}_m+(1-\lambda)\boldsymbol{a}_{\hat{m}}, \quad (15)$$
> $$\widetilde{\boldsymbol{z}}\coloneqq\lambda\boldsymbol{z}_{nm}+(1-\lambda)\boldsymbol{z}_{\hat{n}\hat{m}}, \quad (16)$$
> $$\lambda\sim\mathrm{Beta}(\alpha,\alpha),\alpha>0. \quad (17)$$

Applying the above formulation allows us to handle varying numbers of noisy class labels per instance while considering which class label originates from which annotator. Moreover, we can natively manage even datasets with only one class label for each instance.

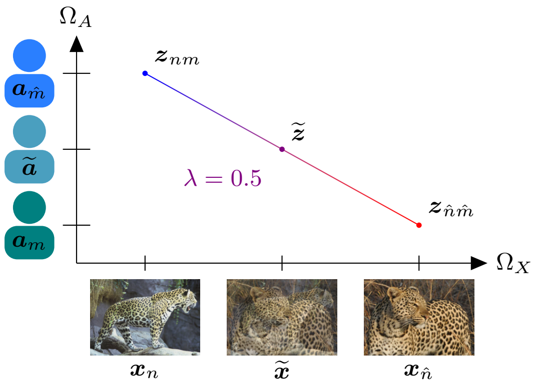

Figure 4. `mixup` extension to multi-annotator classification: We convexly combine class labels from (potentially) different annotators and instances and thus augment data in the instance and annotator feature space.

Figure 4 illustrates our `mixup` extension as data augmentation performed in the instance and annotator feature space. Intuitively, this has two main effects. On the one hand, we simultaneously regularize the classification and annotator model. This is because mixing class labels from different annotators across instances makes it more difficult to memorize which class label an annotator provides for an instance. On the other hand, we improve the generalization by not only linearly interpolating the instance but also the annotator feature space. We demonstrate both effects in our ablation and hyperparameter study (cf. Section 5), which includes an analysis of the α hyperparameter controlling the degree of regularization. For example, defining $\alpha \to 0$ recovers the ERM solution.

4.3 Implementation

Figure 5 summarizes the training with `annot-mix` as a Python code snippet. The design of the classification model's architecture depends on the underlying data modality and task. For example, one may employ a residual network (ResNet) [17] for image data. In contrast, the annotator model must process two vectors as inputs. For this purpose, we use a simple multi-layer perception (MLP) with input concatenation. Our repository provides more implementation details.[1]

```
# Build data loaders for the set M (cf. Eq. (13)).
loaders = zip(loader1, loader2)
# Iterate over the randomly shuffled data from
# both data loaders in batches.
for (x1, a1, z1), (x2, a2, z2) in loaders:
    # Sample mixing coefficient (cf. Eq. (17)).
    lmbda = np.random.beta(alpha, alpha)
    # Perform mixup (cf. Eqs. (14), (15), (16)).
    x = lmbda * x1 + (1-lmbda) * x2
    a = lmbda * a1 + (1-lmbda) * a2
    z = lmbda * z1 + (1-lmbda) * z2
    # Jointly optimize the classification model's
    # parameters θ and the annotator model's
    # parameters π (cf. Eq. (11)).
    optimizer.zero_grad()
    loss = -(z * net(x, a).log()).sum() / len(z)
    loss.backward()
    optimizer.step()
```

Figure 5. Python code snippet for one epoch training with `annot-mix`: The NN architectures are implemented through a PyTorch module `net`, which takes the tensors of mixed instances and annotators as input to minimize the cross-entropy regarding the tensors of mixed noisy class labels.

5 Empirical Evaluation

Our empirical evaluation comprises three parts. First, we explain the basic setup of our experiments. Second, we present the results of an ablation and hyperparameter study. Third, we compare the performance of `annot-mix` to state-of-the-art multi-annotator classification approaches. Code and further details on how to reproduce all experiments are available in our repository.[1]

5.1 Experimental Setup

We design experiments according to the problem setup of Section 2. Table 1 overviews our setup, which we detail in the following.

Datasets: We select our datasets to cover a wide range of real-world settings for obtaining meaningful assessments of the approaches' robustness and performances. Concretely, experiments are performed on eleven real-world datasets across three data modalities: image, tabular, and text. The number of classes ranges from $C = 6$ to $C = 1{,}000$. Five datasets contain noisy class labels from humans, while we simulate annotators providing noisy class labels for the remaining six datasets. The number of annotators ranges from $M = 20$ to $M = 733$. As labels are costly, the average number of provided class labels per instance (approximately ranging from one to four) is considerably lower than the number of annotators. The fraction of false class labels, i.e., noise levels, ranges from low noise (ca. 20 %) to moderate noise (ca. 40 %) to high noise (ca. 80 %) levels.

Annotator Simulation: Due to the costly annotation process, the number of publicly available datasets annotated by multiple error-prone humans is limited. Therefore, we include datasets with simulated annotators. Ideally, the simulated noisy class labels are close to human noisy class labels. To do so, we follow related work [16] and train an individual NN for each annotator. These NNs differ in their training hyperparameters and in the training data they use. Specifically, we train $M = 20$ NNs with different parameter initializations, numbers of training epochs, learning rates, and ratios of randomly sampled training instance-label pairs per class. Then, each NN's predictions serve as the noisy class labels. For example, in a binary classification problem, one NN may train with many instance-label pairs from both classes but only for a few epochs. In contrast, another NN may train for more epochs with fewer instance-label pairs of the positive class. This way, we mimic annotators with different expertise regarding certain classes and regions in the instance feature space. Obviously, we have access to the noisy class label of each simulated annotator for each instance. Yet, we set the average number of class labels per instance to a much lower number (three or one) to account for the limited annotation budget in real applications. Moreover, it is also common for some annotators to provide many labels while other annotators provide very few labels. We address this issue by adopting an existing method [51], where each annotator is assigned an individual probability for annotating an instance. This probability is proportional to the sampled value from a Beta distribution, which we parameterize as $\mathrm{Beta}(1.0, 3.0)$.

Multi-annotator classification approaches: For benchmarking the performance of `annot-mix`, we compare it to eight one-stage multi-annotator classification approaches, which are `crowd-layer` [34], trace-regularized estimation of annotator confusion (`trace-reg`) [40], common noise adaption layers (`conal`) [6], learning from multiple annotators as a union (`union-net`) [45], multi-annotator deep learning (`madl`) [19],

[1] Our GitHub repository is accessible via https://github.com/ies-research/multi-annotator-machine-learning/tree/annot-mix.

geometry-regularized crowdsourcing networks (`geo-reg-w`, `geo-reg-f`) [22], and learning from crowds with annotation reliability (`crowd-ar`) [4]. These one-stage approaches mainly differ regarding their training algorithms and estimation of annotators' performances. While prioritizing one-stage approaches for their reported performance gains [22], we still include basic two-stage approaches to better contextualize the results. Specifically, we employ majority voting (`mv-base`) as a lower baseline and combine vanilla `mixup` [50] with majority voting (`mv-mixup`) and the Dawid-Skene algorithm [8] (`ds-mixup`) as more advanced two-stage approaches. Further, we show the results for training with the true class labels (`true-base`) as an upper baseline.

Evaluation scores: According to our objective in Eq. (1), we assess a classification model with parameters $\boldsymbol{\theta}$ through its empirical classification accuracy on a separate test set $\mathcal{T} \subset \Omega_X \times \Omega_Y$:

$$\texttt{clf-acc}_{\mathcal{T}}(\boldsymbol{\theta}) := \frac{1}{|\mathcal{T}|} \sum_{(\boldsymbol{x}_t, \boldsymbol{y}_t) \in \mathcal{T}} \boldsymbol{y}_t^{\mathrm{T}} \boldsymbol{y}_{\boldsymbol{\theta}}(\boldsymbol{x}_t), \tag{18}$$

where the instance-label pairs in $\mathcal{T}$ are independently sampled from the joint distribution $\Pr(\boldsymbol{x}, \boldsymbol{y})$. Going beyond the standard classification setting, we additionally assess the annotator model with parameters $\boldsymbol{\pi}$. For this purpose, we adopt the idea of evaluating how well the model can predict whether an annotator provides a wrong or correct class label for a certain instance [19]. In case of `annot-mix`, we define a function $p_{\boldsymbol{\theta},\boldsymbol{\pi}} : \Omega_X \times \Omega_A \to [0, 1]$, which outputs the estimated probability of obtaining a correct class label from an annotator $\boldsymbol{a}_m$ for a given instance $\boldsymbol{x}_n$:

$$p_{\boldsymbol{\theta},\boldsymbol{\pi}}(\boldsymbol{x}_n, \boldsymbol{a}_m) := \boldsymbol{p}_{\boldsymbol{\theta}}^{\mathrm{T}}(\boldsymbol{x}_n) \mathrm{diag}\left(\mathbf{P}_{\boldsymbol{\pi}}(\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_n), \boldsymbol{a}_m)\right), \tag{19}$$

where $\mathrm{diag}\left(\mathbf{P}_{\boldsymbol{\pi}}(\boldsymbol{h}_{\boldsymbol{\theta}}(\boldsymbol{x}_n), \boldsymbol{a}_m)\right) \in [0, 1]^C$ denotes the diagonal of the confusion matrix as a column vector. Other one-stage multi-annotator classification approaches similarly provide performance estimates of annotators. Since this estimation task can be interpreted as a binary classification task, we compute the area under the receiver operating characteristic [21] (`perf-auroc`) to assess how well the different approaches can predict annotators' performances. Another evaluation score of interest is the accuracy in predicting the noisy class labels provided by the annotators for the training data:

$$\texttt{annot-acc}_{\mathcal{M}}(\boldsymbol{\theta}, \boldsymbol{\pi}) := \frac{1}{|\mathcal{M}|} \sum_{(\boldsymbol{x}_n, \boldsymbol{a}_m, \boldsymbol{z}_{nm}) \in \mathcal{M}} \boldsymbol{z}_{nm}^{\mathrm{T}} \boldsymbol{z}_{\boldsymbol{\theta},\boldsymbol{\pi}}(\boldsymbol{x}_n, \boldsymbol{a}_m), \tag{20}$$

which allows us to identify overfitting by comparing it to `clf-acc`.

Architectures: We specify architectures to meet the requirements of the respective datasets. For the three tabular datasets `mgc`, `letter`, and `aloi`, we train a simple MLP with two hidden layers of parameters. For the datasets `cifar10h`, `cifar10n`, and `cifar100n`, which consist of 32×32 images, we employ a ResNet18 [17]. The other three image datasets `labelme`, `flowers102`, and `dtd` contain higher-resolution images, so we use DINOv2 [31] as a pre-trained vision transformer (ViT). More concretely, we freeze the feature extraction layers of the ViT-S/14 and train the classification head implemented through an MLP with one hidden layer of parameters. Typical image data augmentations are performed for the six image datasets. An analog procedure is applied to the text datasets `trec6` and `agnews`, with the difference that we use bidirectional encoder representations from transformers (BERT) [10] as a pre-trained architecture.

Training: For all datasets, we employ RAdam [28] as the optimizer and cosine annealing [29] as the learning rate scheduler. Training hyperparameters (cf. Table 1), such as the initial learning rate, the batch size, the number of training epochs, and weight decay, are empirically specified to ensure proper learning and convergence of the `true-base`. We set further hyperparameters specific to a multi-annotator classification approach according to the recommendations of the respective authors. This way, we ensure meaningful and fair comparisons. Moreover, a training, validation, and test set is given for each dataset. If no validation set is provided by the respective data creators, we define a small validation set with true class labels. In this case, the validation size is set either to 100, 500, or 2,000, depending on the number N of training instances. Following related works [36, 51], we use such a validation set to select the model parameters with the highest validation accuracy throughout the training epochs. However, acquiring a validation set with true class labels may be costly in settings with noisy class labels [48]. Thus, we also report the results for the models obtained after the last training epoch. Each experiment is repeated ten times with different parameter initializations. Accordingly, all results refer to means and standard deviations over these ten repetitions.

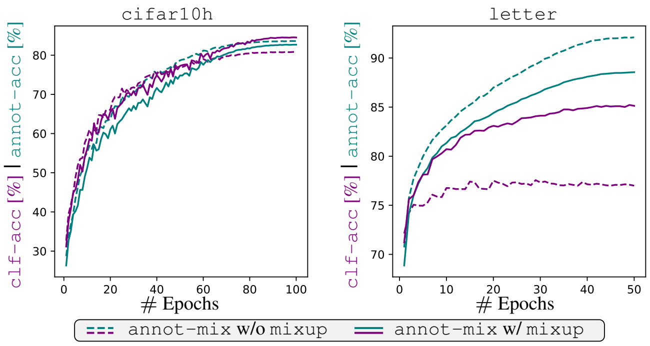

Figure 6. Exemplary learning curves of `annot-mix` with (w/) and without (w/o) `mixup` for the datasets `ciafar10h` and `letter`.

5.2 *Ablation and Hyperparameter Study*

This study analyzes the regularization and generalization effect of our `mixup` extension as a part of `annot-mix`. Further, we study the gain of mixing annotators and class labels across different instances.

Regularization effect: Figure 6 exemplarily depicts the learning curves of `annot-mix` with ($\alpha = 1$, solid line) and without ($\alpha \to 0$, dashed line) our `mixup` extension for the datasets `ciafar10h` and `letter`. The colors distinguish the two evaluation scores `clf-acc` (test set) and `annot-acc` (training set). The observation that the greenish dashed learning curves surpass the greenish solid learning curves demonstrates that training `annot-mix` with our `mixup` extension diminishes the accuracy of predicting noisy class labels assigned by annotators within the training set. In other words, our `mixup` extension makes memorizing the training data more difficult. Yet, the observation that hat the purplish dashed learning curves fall short of the purplish solid learning curves demonstrates that our `mixup` extension boosts the test accuracy. Together, these observations verify our `mixup` extension reduces overfitting to noisy labels.

Generalization effect: Table 2 shows the generalization effect and robustness regarding the hyperparameter α, used for sampling the mixing coefficient λ in Eq. (17). A key observation is that for the vast majority of tested α values and datasets, integrating our `mixup` extension into the training of `annot-mix` improves its generalization performance. Consequently, these performance gains are also robust to some extent regarding the choice of the hyperparameter α. Still,

Table 1. Experimental setup: Column headings indicate the names of the eleven datasets and rows refer to data properties. We denote numbers by prefixing them with the # symbol and indicate averages by $\overline{\#}$. The statistics of `cifar10h` and `cifar10n` refer to annotation subsets of the original datasets.

Setup	`mgc` [35]	`labelme` [34]	`cifar10h` [32]	`cifar10n` [46]	`cifar100n` [46]	`letter` [13]	`flowers102` [30]	`trec6` [27]	`aloi` [15]	`dtd` [7]	`agnews` [53]
General											
data modality	tabular	image	image	image	image	tabular	image	text	tabular	image	text
# training instances	700	1,000	8,621	50,000	50,000	15,500	1,020	4,952	84,400	1,880	118,000
# validation instances	100	500	500	500	500	500	1,020	500	2,000	1,880	2,000
# test instances	200	1,188	49,500	9,500	9,500	4,000	6,149	500	21,600	1,880	7,600
# classes	10	8	10	10	100	26	102	6	1,000	47	4
Annotations											
human annotators	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗
# annotators	44	59	100	733	519	20	20	20	20	20	20
$\overline{\#}$ class labels per instance	4.2	2.5	2.3	1.0	1.0	3.0	3.0	3.0	1.0	1.0	1.0
$\overline{\#}$ class labels per annotator	70	43	200	68	96	2,325	152	743	4,220	94	5,900
% false class labels	44.0	26.0	22.5	40.2	40.2	51.9	67.5	36.9	43.4	76.8	56.8
Training											
architecture	MLP	DINOv2	ResNet18	ResNet18	ResNet18	MLP	DINOv2	BERT	MLP	DINOv2	BERT
pretrained	✗	✓	✗	✗	✗	✗	✓	✓	✗	✓	✓
# epochs	50	50	100	100	100	50	50	50	50	50	50
optimizer	RAdam	RAdam	RAdam	RAdam	RAdam	RAdam	RAdam	RAdam	RAdam	RAdam	RAdam
batch size	64	64	128	128	128	64	64	64	64	64	64
learning rate	1e-2	1e-2	1e-3	1e-3	1e-3	1e-2	1e-2	1e-2	1e-2	1e-2	1e-2
weight decay	0	1e-4	1e-4	1e-4	1e-4	0	1e-4	1e-4	0	1e-4	1e-4

Table 2. Ablation and hyperparameter study of `annot-mix`: **Best** and **second best** results for the `clf-acc` [%] (cf. Eq. (18)) are marked per dataset (row-wise). Numbers right to the datasets (second column) indicate false label fractions [%].

`clf-acc` [%]		w/o `mixup`	w/ `mixup`				idea [54]
		$\alpha \to 0.0$	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 2.0$	$\alpha = 4.0$	$\alpha = 1.0$
Last Epoch							
`mgc`	44.0	**74.5**$_{\pm 1.7}$	**74.2**$_{\pm 1.7}$	73.8$_{\pm 1.1}$	73.3$_{\pm 1.5}$	72.7$_{\pm 1.5}$	72.4$_{\pm 1.5}$
`labelme`	26.0	83.7$_{\pm 2.0}$	85.7$_{\pm 0.6}$	**85.8**$_{\pm 0.6}$	**86.0**$_{\pm 0.4}$	85.4$_{\pm 0.5}$	83.6$_{\pm 0.5}$
`cifar10h`	22.5	80.8$_{\pm 0.3}$	83.8$_{\pm 0.2}$	84.6$_{\pm 0.2}$	**85.0**$_{\pm 0.2}$	**84.9**$_{\pm 0.2}$	80.8$_{\pm 0.2}$
`cifar10n`	40.2	72.7$_{\pm 0.3}$	79.8$_{\pm 0.2}$	82.4$_{\pm 0.5}$	**84.8**$_{\pm 0.2}$	**85.3**$_{\pm 0.2}$	72.7$_{\pm 0.3}$
`cifar100n`	40.2	59.4$_{\pm 0.4}$	63.4$_{\pm 0.4}$	**64.7**$_{\pm 0.4}$	**65.4**$_{\pm 0.3}$	64.6$_{\pm 0.3}$	59.4$_{\pm 0.4}$
`letter`	51.9	76.6$_{\pm 1.8}$	**85.5**$_{\pm 1.2}$	**85.1**$_{\pm 1.3}$	84.7$_{\pm 0.5}$	83.1$_{\pm 0.4}$	82.6$_{\pm 2.0}$
`flowers102`	67.5	88.4$_{\pm 1.1}$	**90.6**$_{\pm 1.2}$	**90.1**$_{\pm 0.9}$	89.8$_{\pm 1.2}$	89.2$_{\pm 1.1}$	89.5$_{\pm 1.0}$
`trec6`	36.9	**93.3**$_{\pm 0.7}$	92.1$_{\pm 0.4}$	**92.3**$_{\pm 0.6}$	91.7$_{\pm 0.7}$	91.6$_{\pm 0.6}$	92.2$_{\pm 0.7}$
`aloi`	43.4	80.7$_{\pm 0.1}$	**83.8**$_{\pm 0.1}$	**83.6**$_{\pm 0.1}$	83.1$_{\pm 0.1}$	82.5$_{\pm 0.1}$	80.7$_{\pm 0.1}$
`dtd`	76.8	52.9$_{\pm 1.1}$	**55.6**$_{\pm 1.7}$	55.2$_{\pm 1.1}$	**55.3**$_{\pm 1.2}$	54.0$_{\pm 1.0}$	52.9$_{\pm 1.1}$
`agnews`	56.8	77.9$_{\pm 4.6}$	83.1$_{\pm 2.3}$	86.2$_{\pm 1.5}$	**86.5**$_{\pm 1.4}$	**86.8**$_{\pm 1.4}$	77.9$_{\pm 4.6}$
Best Epoch							
`mgc`	44.0	72.8$_{\pm 1.8}$	73.0$_{\pm 2.1}$	**73.8**$_{\pm 2.1}$	**73.2**$_{\pm 1.5}$	72.7$_{\pm 2.2}$	71.5$_{\pm 1.7}$
`labelme`	26.0	86.1$_{\pm 0.7}$	**86.8**$_{\pm 0.6}$	**86.5**$_{\pm 0.7}$	86.0$_{\pm 0.4}$	85.6$_{\pm 0.7}$	86.2$_{\pm 0.8}$
`cifar10h`	22.5	80.2$_{\pm 0.6}$	83.4$_{\pm 0.4}$	84.4$_{\pm 0.3}$	**84.5**$_{\pm 0.6}$	**84.7**$_{\pm 0.4}$	80.5$_{\pm 0.6}$
`cifar10n`	40.2	81.0$_{\pm 0.6}$	82.7$_{\pm 0.5}$	83.2$_{\pm 0.3}$	**84.7**$_{\pm 0.3}$	**85.4**$_{\pm 0.2}$	81.0$_{\pm 0.6}$
`cifar100n`	40.2	58.0$_{\pm 1.4}$	62.7$_{\pm 0.5}$	**64.1**$_{\pm 0.6}$	**64.7**$_{\pm 0.6}$	64.0$_{\pm 0.6}$	58.0$_{\pm 1.4}$
`letter`	51.9	77.5$_{\pm 1.7}$	**85.1**$_{\pm 1.2}$	**84.9**$_{\pm 1.6}$	84.8$_{\pm 0.5}$	83.1$_{\pm 0.7}$	82.4$_{\pm 2.5}$
`flowers102`	67.5	88.4$_{\pm 1.2}$	**90.5**$_{\pm 1.2}$	**90.2**$_{\pm 0.9}$	89.8$_{\pm 1.2}$	89.1$_{\pm 1.1}$	89.5$_{\pm 1.1}$
`trec6`	36.9	**91.8**$_{\pm 1.2}$	**92.0**$_{\pm 0.9}$	91.4$_{\pm 1.0}$	91.3$_{\pm 0.9}$	91.0$_{\pm 1.0}$	91.6$_{\pm 0.8}$
`aloi`	43.4	80.4$_{\pm 0.3}$	**83.7**$_{\pm 0.2}$	**83.6**$_{\pm 0.2}$	83.0$_{\pm 0.1}$	82.4$_{\pm 0.2}$	80.4$_{\pm 0.3}$
`dtd`	76.8	53.2$_{\pm 1.0}$	**55.8**$_{\pm 1.5}$	55.1$_{\pm 0.9}$	**55.4**$_{\pm 1.1}$	53.8$_{\pm 1.0}$	53.2$_{\pm 1.0}$
`agnews`	56.8	82.4$_{\pm 3.2}$	86.1$_{\pm 1.4}$	87.0$_{\pm 0.7}$	**87.4**$_{\pm 0.8}$	**87.4**$_{\pm 0.6}$	82.4$_{\pm 3.2}$

there are a few noteworthy differences, e.g., larger α values tend to perform better when training randomly initialized ResNets, whereas smaller (non-zero) α values tend to provide larger gains when training pretrained models and MLPs. Despite these differences, we fix $\alpha = 1$ for all subsequent empirical evaluations. On the one hand, this parameterization corresponds to the common choice of a uniform distribution over the mixing coefficient λ [50]. On the other hand, choosing such a default value allows for a fair comparison with the other multi-annotator classification approaches.

Mixing triples with different instances: Inspired by the idea of Zhang et al. [54], we modify our `mixup` extension to only combine two triples $(\boldsymbol{x}_n, \boldsymbol{a}_m, \boldsymbol{z}_{nm}), (\boldsymbol{x}_{\hat{n}}, \boldsymbol{a}_{\hat{m}}, \boldsymbol{z}_{\hat{n}\hat{m}}) \in \mathcal{M}$, if and only if both instances are equal, i.e., $\boldsymbol{x}_n = \boldsymbol{x}_{\hat{n}}$. Evaluating this modified `mixup` extension while keeping the rest of `annot-mix` unchanged allows us to study the benefit of mixing triples containing different instances. This modification of `annot-mix` is not equivalent to the approach of Zhang et al. [54], and we refer to Section 3 for further differences. Table 2 presents the results of `annot-mix` with this modification in its last column. For the datasets with one class label per instance, the results are identical to the training w/o `mixup`, as mixing only happens if multiple class labels per instance are available. The performance gains compared to training w/o `mixup` are noteworthy for the datasets `letter` and `flowers102`, while no substantial gains are observed for the other datasets. Furthermore, the performance results across almost all tested datasets fall short of those achieved by our original `mixup` extension as part of `annot-mix`. This observation highlights the importance and broader applicability of mixing triples containing different instances.

5.3 Benchmark Study

Classification models: Table 3 presents the results for comparing the classification models' performances trained by the different approaches per dataset. As expected, training with the true class labels, i.e., `true-base`, leads to the best results across all datasets. Comparing the performances of this upper baseline to `mv-base` as the lower baseline, we clearly observe the negative impact of the noisy class labels. For example, the performance gap between the lower and upper baseline is about 40% for the dataset `dtd`, which contains the highest fraction of false class labels. The approach `mv-mixup` strongly reduces this performance gap for almost all datasets and thus confirms the benefit of vanilla `mixup` in combination with two-stage approaches. For the dataset with multiple class labels per instance, training with `ds-mixup` yields additional performance gains, establishing it as a strong two-stage competitor to the one-stage approaches. If we now also consider the results of these one-stage approaches, we recognize that `annot-mix` is the only approach outperforming the two-stage approaches for almost each data set. The other one-stage approaches often perform inferiorly on datasets with many classes. For example, except `annot-mix`, all one-stage approaches are worse than `mv-mixup` for the data set `aloi` with $C = 1{,}000$ classes. Further, `annot-mix` outperforms its competitors by considerable margins for four of the five datasets annotated by humans. Comparing the classification models' performances after the last epoch and after the best epoch, selected via a validation set during training, we observe inconsistent improvements in this model selection. This is mainly due to the small size of the validation sets. However, this also shows another advantage of `annot-mix`, which performs better than its competitors on ten out of eleven data sets when no expensive validation sets with clean labels are available.

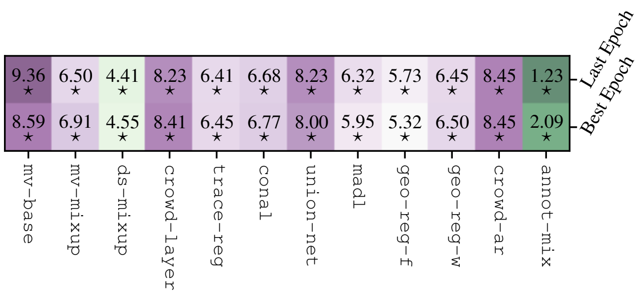

Figure 7. Benchmark study: Numbers refer to the approaches' mean ranks across the datasets in Table 3. Lower values mean better ranks. A star ($\star$) marks that `annot-mix` performs significantly superior to its respective competitor.

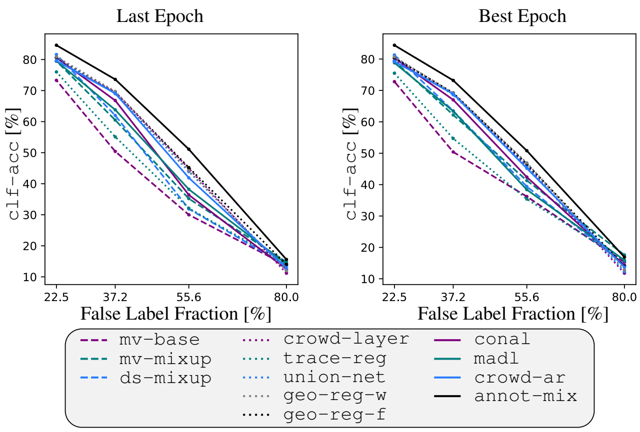

Figure 8. Benchmark study: Each curve shows the `clf-acc` [%] (cf. Eq. (18)) of an approach across four false label fractions [%] for `cifar10h`.

For a more compact presentation of the results in Table 3, we further compute each approach's rank per dataset and report their means in Fig. 7. Moreover, we evaluate statistical significance at the level of 0.05 by following a common test protocol [9]. Concretely, we perform a Friedman test [14] as an omnibus test with the null hypothesis that all approaches perform the same and observed performance differences are due to randomness. If this null hypothesis is rejected, we proceed with Dunn's post-hoc test [11] for pairwise multiple comparisons between `annot-mix` and each of its competitors. Thereby, we employ Holm's step-down procedure [20] to control for the family-wise error rate. This test protocol is applied to the classification model's performances after the last and the best epoch. The results demonstrate that `annot-mix` significantly outperforms each competitor. Additionally, it is noteworthy that `ds-mixup` demonstrates greater robustness across datasets as a two-stage approach compared to its one-stage competitors (with the exception of `annot-mix`).

Figure 8 concludes our analysis by showing the approaches' test accuracies for four different variants of the `cifar10h` dataset as curves. These variants include only class labels from the original dataset's $M = 10$, $M = 20$, $M = 40$, and $M = 100$ least accurate annotators, inducing four different fractions of false class labels. As expected, each approach's test accuracy decreases with an increasing false label fraction. Further, the lower baseline `mv-base` performs worst, as shown by the low-lying dashed purplish curve, whereas the solid black curve of `annot-mix` mostly remains above the other curves, demonstrating its superior performance across various false label fractions.

Annotator models: Table 4 presents the results for comparing the annotator models' performances. Intuitively, a high score implies that the corresponding annotator model can accurately predict whether an annotator will provide a correct or false class label for a given instance. We include only the results for the datasets with simulated annotators because the test sets of the other datasets were not annotated by error-prone humans. Further, only the related approaches that train an annotator model are considered. For the results after the last and best epoch, we observe that `annot-mix` performs best on four of the six datasets while providing competitive results for the other two datasets. As a result, our approach has the potential to be used in applications where it is important to obtain accurate predictions of the annotators' performances, e.g., when selecting the best annotator to provide class labels in an active learning setting [18].

6 Conclusion and Outlook

In this article, we proposed our approach `annot-mix` addressing the practical challenge of learning from noisy class labels provided by multiple annotators. It maximizes the marginal likelihood of the observed noisy class labels during the joint training of a classification and an annotator model, effectively separating the noise from the true labels. An essential property of our approach is the integration of our novel `mixup` extension, which convexly combines triples of instances, annotators, and noisy class labels. This data augmentation and regularization technique makes memorizing individual noisy class labels more difficult and thus reduces the risk of overfitting. An extensive empirical evaluation with eleven datasets of three data modalities demonstrated that `annot-mix` significantly outperforms current state-of-the-art multi-annotator classification approaches.

A future research direction for our work is to adopt ideas of other `mixup` extensions such as `cut-mix` [49] for image data or `manifold-mixup` [44] for combining hidden states of both, the classification and annotator model. Moreover, throughout this article, we used only one-hot encoded representations of the annotators since no datasets with metadata about the annotators exist. Accordingly, collecting such datasets and making them publicly available to the research community would allow us to evaluate the benefit of such metadata. The annotator model's results of `annot-mix` suggest its potential application to query the most accurate annotators for the most informative instances in active learning settings [18]. A general challenge of multi-annotator classification is reliance on small validation sets, which can render model selection unreliable. Future work could focus on developing model selection processes without necessitating large validation sets [48]. Finally, the extension of `annot-mix` toward related task types, such as semantic segmentation, by adjusting the classification and annotator model architectures would further broaden its practical use.

7 Ethical Statement

We confirm that our research refrains from any experimentation with humans. Yet, we emphasize the issue that human annotators, particularly crowdworkers, often endure difficult working conditions [2], e.g., minimum job security and low salaries, despite their essential contributions to advancing machine learning research and applications. Although `annot-mix` allows assessing annotators' performances, we recommend adhering to strict guidelines to avoid unjustified discrimination against annotators. Furthermore, we emphasize our work's empirical nature and, therefore, suggest a thorough empirical evaluation before its application to safety-critical domains.

Table 3. Benchmark study: **Best**, **second best**, and worse than `mv-base` results of the `clf-acc` [%] (cf. Eq. (18)) are marked per dataset (column-wise) while excluding results of `true-base`. Numbers below datasets (second row) show false label fractions [%]. The results of `mv-mixup` and `ds-mixup` are identical for the datasets with only one class label per instance since the Dawid-Skene algorithm [8] reduces to majority voting in these cases.

clf-acc [%]	mgc	labelme	cifar10h	cifar10n	cifar100n	letter	flowers102	trec6	aloi	dtd	agnews
	44.0	26.0	22.5	40.2	40.2	51.9	67.5	36.9	43.4	76.8	56.8
Last Epoch											
true-base	79.6±0.9	93.9±0.3	85.2±0.2	94.0±0.2	74.5±0.3	98.0±0.1	99.5±0.0	93.7±0.6	95.7±0.1	78.1±0.3	92.9±0.1
mv-base	66.1±2.8	80.5±1.1	73.3±0.2	63.8±0.6	51.9±0.3	75.6±0.7	68.0±1.4	86.1±1.1	71.9±0.3	35.6±0.9	74.3±0.5
mv-mixup	68.2±2.1	82.8±0.7	79.5±0.3	81.3±0.4	**60.0**±0.3	81.3±0.4	71.7±1.3	89.0±0.9	**81.3**±0.2	43.0±1.1	74.4±0.7
ds-mixup	68.9±1.5	85.3±0.4	**81.6**±0.2			**83.9**±0.4	78.9±0.3	91.0±0.5			
crowd-layer	69.8±1.0	**85.7**±0.5	79.5±0.5	77.9±0.4	4.8±0.6	56.8±2.3	36.8±2.2	91.1±0.6	0.9±0.3	31.3±1.1	85.6±0.3
trace-reg	66.4±1.4	82.6±0.6	76.0±0.4	65.1±0.5	53.7±0.5	82.7±0.3	76.2±0.5	**92.0**±0.5	60.4±0.5	36.6±0.5	**86.7**±0.2
conal	69.0±0.9	83.7±0.4	80.6±0.2	77.9±0.4	27.4±1.4	82.5±1.4	52.8±3.1	90.1±0.7	21.3±1.4	40.9±1.8	76.1±0.4
union-net	68.6±1.0	85.2±0.3	80.5±0.4	**81.4**±0.5	1.3±0.6	66.3±2.0	43.1±3.0	90.1±0.3	0.9±0.2	30.9±1.8	86.2±0.3
madl	**72.0**±2.0	82.5±0.8	79.5±0.5	76.9±0.4	42.8±7.4	69.1±4.6	**85.0**±1.9	91.1±0.7	79.2±0.3	**47.2**±1.8	76.0±10.8
geo-reg-f	70.2±0.7	85.4±0.5	80.7±0.4	80.5±0.3	8.1±0.9	82.4±1.9	44.5±3.3	91.8±0.3	1.6±0.3	35.2±1.5	**86.7**±0.2
geo-reg-w	70.1±0.9	85.5±0.5	80.9±0.4	79.8±0.2	8.1±0.9	73.9±3.0	44.5±3.3	91.9±0.6	1.7±0.3	34.6±1.3	82.1±9.0
crowd-ar	69.0±1.8	84.6±0.7	79.6±0.2	80.5±0.5	1.0±0.0	78.1±2.2	48.2±2.5	89.4±0.4	0.1±0.1	39.3±1.5	72.4±4.8
annot-mix	**73.8**±1.1	**85.8**±0.6	**84.6**±0.2	**82.4**±0.5	**64.7**±0.4	**85.1**±1.3	**90.1**±0.9	**92.3**±0.6	**83.6**±0.1	**55.2**±1.1	86.2±1.2
Best Epoch											
true-base	78.9±1.1	93.8±0.5	84.8±0.5	93.5±0.7	74.4±0.3	97.6±0.6	99.4±0.1	93.3±0.6	95.7±0.1	77.6±0.5	92.8±0.2
mv-base	66.8±2.6	85.5±0.8	72.8±0.8	79.1±0.7	53.2±1.1	78.4±0.9	71.5±1.2	87.4±0.6	74.9±0.4	46.9±0.9	77.1±0.8
mv-mixup	67.5±3.3	85.1±1.2	79.4±0.4	**82.8**±0.6	**60.2**±0.4	81.3±0.7	71.8±1.3	88.2±1.0	**81.2**±0.3	46.2±1.4	76.6±1.4
ds-mixup	69.0±2.0	87.2±1.2	**81.3**±0.5			**83.4**±0.5	79.5±0.6	90.8±0.9			
crowd-layer	69.0±1.1	87.3±0.5	79.4±0.5	81.9±0.5	4.5±0.5	57.2±2.1	36.9±2.5	90.7±0.8	1.0±0.3	31.9±1.2	86.2±0.6
trace-reg	67.8±2.1	85.8±0.7	75.5±0.6	79.2±0.8	54.3±0.8	82.8±0.5	77.9±0.4	91.4±0.8	64.5±0.9	46.9±1.0	86.2±0.6
conal	69.2±2.0	87.1±0.6	80.3±0.4	79.9±0.6	27.1±1.6	82.8±1.3	52.8±3.0	90.0±0.4	21.2±1.4	41.8±1.5	79.3±1.1
union-net	68.5±1.7	**87.5**±0.5	80.2±0.6	82.0±0.4	4.0±0.6	66.8±1.9	44.0±3.3	89.8±0.6	0.9±0.2	33.6±1.0	**87.1**±0.4
madl	**72.4**±1.8	86.5±0.8	78.8±1.2	80.5±0.9	42.7±7.3	71.2±4.0	**85.1**±1.9	91.1±0.8	79.0±0.4	**47.7**±1.3	78.0±10.0
geo-reg-f	70.4±0.8	**87.4**±0.6	80.4±0.7	81.9±0.5	7.9±1.0	82.4±1.5	44.9±3.3	**91.5**±0.9	1.6±0.3	35.9±1.3	**87.4**±0.4
geo-reg-w	69.8±1.1	87.3±0.4	80.7±0.5	81.5±0.5	7.9±1.1	74.2±2.4	44.9±3.3	**91.6**±1.0	1.6±0.3	35.8±1.4	83.1±8.1
crowd-ar	70.4±2.0	87.3±0.5	79.2±0.7	80.4±0.4	4.3±0.8	77.8±2.2	48.2±2.4	89.1±1.0	1.0±0.4	40.1±1.5	76.3±2.7
annot-mix	**73.8**±2.1	86.5±0.7	**84.4**±0.3	**83.2**±0.8	**64.1**±0.6	**84.9**±1.6	**90.0**±0.9	91.4±1.0	**83.5**±0.1	**55.1**±0.9	87.0±0.7

Table 4. Benchmark study: **Best** and **second best** results of the `perf-auroc` [%] are marked per dataset (row-wise). Numbers right to datasets (second column) show false label fractions [%].

perf-auroc [%]		crowd-layer	trace-reg	conal	union-net	madl	geo-reg-f	geo-reg-w	crowd-ar	annot-mix
Last Epoch										
letter	51.9	78.0±1.5	87.5±0.0	63.0±0.3	86.6±1.0	84.2±2.4	**91.9**±0.9	87.3±1.2	61.9±0.6	**93.1**±0.7
flowers102	67.5	67.2±0.6	77.8±0.8	61.0±0.4	69.5±1.0	**86.0**±0.8	68.7±0.9	68.7±0.9	55.4±1.1	**90.4**±0.3
trec6	36.9	**95.5**±0.1	93.8±0.1	56.3±0.2	**94.7**±0.1	94.6±0.3	93.9±0.1	93.8±0.1	56.3±1.4	94.2±0.1
aloi	43.4	48.1±1.5	70.8±0.2	52.2±0.2	56.2±1.2	**85.9**±0.4	55.4±1.5	55.6±1.5	50.5±4.4	**88.7**±0.1
dtd	76.8	65.2±1.2	55.4±0.2	58.0±0.3	65.9±1.3	**75.4**±1.4	65.8±0.8	66.1±0.7	56.9±0.9	**81.6**±0.6
agnews	56.8	**93.0**±0.1	92.1±0.1	61.7±0.1	**93.0**±0.1	88.1±7.5	92.2±0.1	89.4±5.5	63.4±1.3	92.7±0.5
Best Epoch										
letter	51.9	79.9±1.5	87.8±0.3	62.8±0.2	86.7±0.9	83.8±3.6	**91.4**±1.0	87.0±1.3	62.0±0.5	**93.0**±0.7
flowers102	67.5	65.9±3.5	78.2±0.4	60.9±0.4	69.7±1.1	**85.9**±0.9	69.0±0.9	69.0±0.9	54.8±1.5	**90.4**±0.2
trec6	36.9	**95.4**±0.2	93.7±0.1	56.5±0.4	**94.6**±0.2	94.5±0.3	93.8±0.2	93.7±0.1	56.3±1.4	94.0±0.3
aloi	43.4	54.1±2.1	63.9±1.1	52.1±0.3	56.1±1.3	**86.2**±0.5	55.1±1.7	55.3±1.7	50.8±4.0	**88.7**±0.1
dtd	76.8	62.6±1.9	57.9±0.5	58.5±0.4	61.6±1.2	**74.0**±6.0	64.0±1.4	63.6±1.8	56.0±1.3	**81.5**±0.6
agnews	56.8	**93.0**±0.2	92.4±0.1	61.3±0.8	**93.1**±0.2	89.2±6.3	92.4±0.1	89.7±5.3	63.1±1.3	92.7±0.4

Acknowledgements

This work was funded by the ALDeep project through the University of Kassel (grant number: P/681). Moreover, we thank Lukas Rauch for his insightful comments and discussions, which further improved this article. Finally, we acknowledge the usage of the public domain animal images of the jaguar (credit: Hollingsworth, John and Karen, USFWS) and leopard (credit: USFWS) in Figs. 1, 2, 3, and 4.

References

[1] G. Algan and I. Ulusoy. Image classification with deep learning in the presence of noisy labels: A survey. *Knowl. Based Syst.*, 215:106771, 2021.

[2] S. S. Bhatti, X. Gao, and G. Chen. General framework, opportunities and challenges for crowdsourcing techniques: A comprehensive survey. *J. Syst. Softw.*, 167:110611, 2020.

[3] E. Breck, Y. Choi, and C. Cardie. Identifying expressions of opinion in context. In *Int. Joint Conf. Artif. Intell.*, pages 2683–2688, 2007.

[4] Z. Cao, E. Chen, Y. Huang, S. Shen, and Z. Huang. Learning from Crowds with Annotation Reliability. In *Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pages 2103–2107, 2023.

[5] Z. Chen, L. Jiang, and C. Li. Label augmented and weighted majority voting for crowdsourcing. *Inf. Sci.*, 606:397–409, 2022.

[6] Z. Chu, J. Ma, and H. Wang. Learning from Crowds by Modeling Common Confusions. In *AAAI Conf. Artif. Intell.*, pages 5832–5840, 2021.

[7] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing Textures in the Wild. In *IEEE / CVF Conf. Comput. Vis. Pattern Recognit.*, pages 3606–3613, 2014.

[8] A. P. Dawid and A. M. Skene. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *J. R. Stat. Soc.*, 28(1): 20–28, 1979.

[9] J. Demšar. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.*, 7:1–30, 2006.

[10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Annu. Conf. N. Am. Chapter Assoc. Comput. Linguist.*, pages 4171–4186, 2019.

[11] O. J. Dunn. Multiple Comparisons among Means. *J. Am. Stat. Assoc.*, 56(293):52–64, 1961.

[12] B. Frénay and M. Verleysen. Classification in the Presence of Label Noise: a Survey. *IEEE Trans. Neural Netw. Learn. Syst.*, 25(5):845–869, 2013.

[13] P. W. Frey and D. J. Slate. Letter recognition using Holland-style adaptive classifiers. *Mach. Learn.*, 6(2):161–182, 1991.

[14] M. Friedman. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *J. Am. Stat. Assoc.*, 32(200):675–701, 1937.

[15] J.-M. Geusebroek, G. J. Burghouts, and A. W. Smeulders. The Amsterdam Library of Object Images. *Int. J. Comput. Vis.*, 61:103–112, 2005.

[16] K. Gu, X. Masotto, V. Bachani, B. Lakshminarayanan, J. Nikodem, and D. Yin. An instance-dependent simulation framework for learning with label noise. *Mach. Learn.*, 112(6):1871–1896, 2023.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE / CVF Conf. Comput. Vis. Pattern Recognit.*, pages 770–778, 2016.

[18] M. Herde, D. Huseljic, B. Sick, and A. Calma. A Survey on Cost Types, Interaction Schemes, and Annotator Performance Models in Selection Algorithms for Active Learning in Classification. *IEEE Access*, 9:166970–166989, 2021.

[19] M. Herde, D. Huseljic, and B. Sick. Multi-annotator Deep Learning: A Probabilistic Framework for Classification. *Trans. Mach. Learn. Res.*, 2023.

[20] S. Holm. A Simple Sequentially Rejective Multiple Test Procedure. *Scand. J. Stat.*, pages 65–70, 1979.

[21] J. Huang and C. X. Ling. Using AUC and Accuracy in Evaluating Learning Algorithms. *IEEE Trans. Knowl. Data Eng.*, 17(3):299–310, 2005.

[22] S. Ibrahim, T. Nguyen, and X. Fu. Deep Learning From Crowdsourced Labels: Coupled Cross-Entropy Minimization, Identifiability, and Regularization. In *Int. Conf. Learn. Represent.*, 2023.

[23] L. Jiang, H. Zhang, F. Tao, and C. Li. Learning From Crowds With Multiple Noisy Label Distribution Propagation. *IEEE Trans. Neural Netw. Learn. Syst.*, 33(11):6558–6568, 2021.

[24] A. Khetan, Z. C. Lipton, and A. Anandkumar. Learning From Noisy Singly-labeled Data. In *Int. Conf. Learn. Represent.*, 2018.

[25] A. Krogh and J. Hertz. A Simple Weight Decay Can Improve Generalization. In *Adv. Neural Inf. Process. Syst.*, volume 4, 1991.

[26] J. Li, H. Sun, and J. Li. Beyond confusion matrix: learning from multiple annotators with awareness of instance features. *Mach. Learn.*, pages 1–23, 2022.

[27] X. Li and D. Roth. Learning Question Classifiers. In *Int. Conf. Comput. Linguist.*, 2002.

[28] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the Variance of the Adaptive Learning Rate and Beyond. In *Int. Conf. Learn. Represent.*, 2019.

[29] I. Loshchilov and F. Hutter. SGDR: Stochastic Gradient Descent with Warm Restarts. In *Int. Conf. Learn. Represent.*, 2017.

[30] M.-E. Nilsback and A. Zisserman. Automated Flower Classification over a Large Number of Classes. In *Indian Conf. Comput. Vis., Gr. & Image*, pages 722–729, 2008.

[31] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *Trans. Mach. Learn. Res.*, 2023.

[32] J. C. Peterson, R. M. Battleday, T. L. Griffiths, and O. Russakovsky. Human Uncertainty Makes Classification More Robust. In *IEEE/CVF Int. Conf. Comput. Vis.*, pages 9617–9626, 2019.

[33] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from Crowds. *J. Mach. Learn. Res.*, 11(4):1297–1322, 2010.

[34] F. Rodrigues and F. Pereira. Deep Learning from Crowds. In *AAAI Conf. Artif. Intell.*, pages 1611–1618, 2018.

[35] F. Rodrigues, F. Pereira, and B. Ribeiro. Learning from multiple annotators: Distinguishing good from random labelers. *Pattern Recognit. Lett.*, 34(12):1428–1436, 2013.

[36] S. Rühling Cachay, B. Boecking, and A. Dubrawski. End-to-End Weak Supervision. In *Adv. Neural Inf. Process. Syst.*, 2021.

[37] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee. Learning From Noisy Labels With Deep Neural Networks: A Survey. *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

[38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, 2014.

[39] L. Sun, C. Xia, W. Yin, T. Liang, S. Y. Philip, and L. He. Mixup-Transformer: Dynamic Data Augmentation for NLP Tasks. In *Int. Conf. Comput. Linguist.*, pages 3436–3440, 2020.

[40] R. Tanno, A. Saeedi, S. Sankaranarayanan, D. C. Alexander, and N. Silberman. Learning from Noisy Labels by Regularized Estimation of Annotator Confusion. In *IEEE / CVF Conf. Comput. Vis. Pattern Recognit.*, pages 11244–11253, 2019.

[41] T. Tian and J. Zhu. Max-Margin Majority Voting for Learning from Crowds. In *Adv. Neural Inf. Process. Syst.*, 2015.

[42] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[43] J. W. Vaughan. Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research. *J. Mach. Learn. Res.*, 18 (193):1–46, 2018.

[44] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio. Manifold Mixup: Better Representations by Interpolating Hidden States. In *Int. Conf. Mach. Learn.*, pages 6438–6447, 2019.

[45] H. Wei, R. Xie, L. Feng, B. Han, and B. An. Deep Learning From Multiple Noisy Annotators as A Union. *IEEE Trans. Neural Netw. Learn. Syst.*, 2022.

[46] J. Wei, Z. Zhu, H. Cheng, T. Liu, G. Niu, and Y. Liu. Learning with Noisy Labels Revisited: A Study Using Real-World Human Annotations. In *Int. Conf. Learn. Represent.*, 2021.

[47] J. Yang, T. Drake, A. Damianou, and Y. Maarek. Leveraging Crowdsourcing Data for Deep Active Learning an Application: Learning Intents in Alexa. In *Int. World Wide Web Conf.*, pages 23–32, 2018.

[48] S. Yuan, L. Feng, and T. Liu. Early Stopping Against Label Noise Without Validation Data. In *Int. Conf. Learn. Represent.*, 2024.

[49] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *IEEE / CVF Conf. Comput. Vis. Pattern Recognit.*, pages 6023–6032, 2019.

[50] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond Empirical Risk Minimization. In *Int. Conf. Learn. Represent.*, 2018.

[51] H. Zhang, S. Li, D. Zeng, C. Yan, and S. Ge. Coupled Confusion Correction: Learning from Crowds with Sparse Annotations. In *AAAI Conf. Artif. Intell.*, 2024.

[52] L. Zhang, R. Tanno, M. Xu, Y. Huang, K. Bronik, C. Jin, J. Jacob, Y. Zheng, L. Shao, O. Ciccarelli, et al. Learning from Multiple Annotators for Medical Image Segmentation. *Pattern Recognit.*, page 109400, 2023.

[53] X. Zhang, J. Zhao, and Y. LeCun. Character-level Convolutional Networks for Text Classification. In *Adv. Neural Inf. Process. Syst.*, 2015.

[54] X. Zhang, G. Xu, Y. Sun, M. Zhang, X. Wang, and M. Zhang. Identifying Chinese Opinion Expressions with Extremely-Noisy Crowdsourcing Annotations. In *Annu. Meeting. Assoc. Comput. Linguist.*, pages 2801–2813, 2022.