A GOOD REPRESENTATION DETECTS NOISY LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Label noise is pervasive in real-world datasets, which encodes wrong correlation patterns and impairs the generalization of deep neural networks (DNNs). It is critical to find efficient ways to detect the corrupted patterns. Current methods primarily focus on designing robust training techniques to prevent DNNs from memorizing corrupted patterns. This approach has two outstanding caveats: 1) applying this approach to each individual dataset would often require customized training processes; 2) as long as the model is trained with noisy supervisions, overfitting to corrupted patterns is often hard to avoid, leading to performance drop in detection. In this paper, given good representations, we propose a universally applicable and training-free solution to detect noisy labels. Intuitively, good representations help define "neighbors" of each training instance, and closer instances are more likely to share the same clean label. Based on the neighborhood information, we propose two methods: the first one uses "local voting" via checking the noisy label consensuses of nearby representations. The second one is a ranking-based approach that scores each instance and filters out a guaranteed number of instances that are likely to be corrupted, again using only representations. Given good (but possibly imperfect) representations that are commonly available in practice, we theoretically analyze how they affect the local voting and provide guidelines for tuning neighborhood size. We also prove the worst-case error bound for the ranking-based method. Experiments with both synthetic and real-world label noise demonstrate our training-free solutions are consistently and significantly improving over most of the training-based baselines.

1 INTRODUCTION

The generalization of deep neural networks (DNNs) depends on the quality and the quantity of the data. Nonetheless, in practice real-world datasets often contain label noise that challenges the above assumption (Krizhevsky et al., 2012; Zhang et al., 2017; Agarwal et al., 2016). Employing human workers to cleaning annotations is one reliable way to improve the label quality, but it is too expensive and time-consuming for a large-scale dataset. One promising way to automatically cleaning up label errors is to first algorithmically detect possible label errors from a large-scale dataset (Cheng et al., 2021; Northcutt et al., 2021a; Pruthi et al., 2020; Bahri et al., 2020), and then correct them using either algorithm or crowdsourcing (Northcutt et al., 2021b).

Almost all the algorithmic detection approaches focus on designing customized training processes to learn with noisy labels, where the idea is to train DNNs with noisy supervisions and then make decisions based on the output (Northcutt et al., 2021a) or gradients (Pruthi et al., 2020) of the last logit layer of the trained model. The high-level intuition of these methods is the memorization effects (Han et al., 2020), i.e., instances with label errors, a.k.a., corrupted instances, tend to be harder to be learned by DNNs than clean instances (Xia et al., 2021; Liu et al., 2020). By setting appropriate hyperparameters to utilize the memorization effect, corrupted instances could be identified.

The above methods suffer from two major limitations: 1) the customized training processes are task-specific and may require fine-tuning hyperparameters for different datasets/noise; 2) as long as the model is trained with noisy supervisions, the memorization of corrupted instances exists. The model will "subjectively" and wrongly treat the memorized/overfitted corrupted instances as clean. For example, some low-frequency/rare clean instances may be harder to memorize than high-frequency/common corrupted instances. Memorizing these corrupted instances lead to unexpected and disparate impacts (Liu, 2021). One way to avoid memorizing/overfitting is to drop the depen-

dency on training using the noisy supervisions, which motivates us to design a *training-free* method to find label errors. Intuitively, we can carefully use the information from nearby representations to infer whether one instance is corrupted or not. This observation inspired our solution of using a good representation to first identify "neighbor" instances for each training example, whose noisy labels jointly would help us perform the detection. Note the representation extractor could be fully independent of noisy supervisions by referring to unsupervised learning or self-supervised learning, or even adapting from other tasks.

Our training-free method enables more possibilities beyond a better detection result. For example, the concerns of the required assumptions and hyperparameter tuning in those training-based methods will now be released due to our training-free property. The complexity of such a solution will also be much lower, again due to the removal of the possibly involved training processes. This light detection solution also has the potential to serve as a pre-processing module to prepare data for other sophisticated tasks (e.g., semi-supervised learning (Xie et al., 2019; Berthelot et al., 2019)).

Our main contributions are: 1) *New perspective:* Different from current methods that train customized models on noisy datasets, we proposed a training-free solution to efficiently detect noisy labels given good representations. We believe this is the first attempt of the same kind to the best of our knowledge. 2) *Efficient algorithms:* Based on the neighborhood information, we propose two methods: a voting-based local detection method that only requires checking the noisy label consensuses of nearby representations, and a ranking-based global detection method that scores each instance by its likelihood of being clean and filters out a guaranteed percentage of instances with low scores as corrupted ones. 3) *Theoretical analyses:* We theoretically analyze how a good representation (but possibly imperfect in practice) affects the local voting and provide guidelines for tuning neighborhood size. We also prove the worst-case error bound for the ranking-based method. 4) *Numerical findings:* Our numerical experiments show three important messages: in label noise detection, i) training with noisy supervisions may not be necessary; ii) representation layers tend to be more useful than the logit layers; iii) representations from other tasks or domains are helpful.

1.1 Related Works

Learning with noisy labels There are many other works that can detect corrupted instances (a.k.a. sample selection) in the literature, e.g., (Jiang et al., 2018; Han et al., 2018; Yu et al., 2019; Yao et al., 2020; Wei et al., 2020; Jiang et al., 2020; Zhang et al., 2021), and its combination with semi-supervised learning (Nguyen et al., 2019; Li et al., 2020b; Cheng et al., 2021). Another line of works focus on designing robust loss functions to mitigate the effect of label noise, such as numerical methods (Ghosh et al., 2017; Zhang & Sabuncu, 2018; Gong et al., 2018; Amid et al., 2019; Wang et al., 2019; Shu et al., 2020) and statistical methods (Natarajan et al., 2013; Liu & Tao, 2015; Patrini et al., 2017; Liu & Guo, 2020; Xia et al., 2019; 2021; Wei & Liu, 2021; Zhu et al., 2021a; Li et al., 2021; Liu et al., 2020). All these methods require training DNNs with noisy supervisions.

Pre-training or Self-supervised Learning Training DNNs from scratch may be time-consuming (Goyal et al., 2017). A common strategy to alleviate this issue is initializing DNN with a pre-trained model when training on new tasks (Krizhevsky et al., 2012; Zeiler & Fergus, 2014; Sermanet et al., 2013). The pre-trained model can be used for down-stream tasks due to the great transfer-ability of DNNs (Tan et al., 2018). Popular pre-trained models include BERT (Devlin et al., 2018) for language tasks or CLIP (Radford et al., 2021) for vision-language tasks. Recently, unsupervised learning (Ji et al., 2019) and self-supervised learning (Jaiswal et al., 2021; Liu et al., 2021; He et al., 2020; Chen et al., 2020) also exhibit great power when fine-tuning DNN on down-stream tasks.

Label aggregation Our work is also relevant to the literature of crowdsourcing that focuses on label aggregation (to clean the labels)(Liu et al., 2012; Karger et al., 2011; 2013; Liu & Liu, 2015; Zhang et al., 2014). Most of these works can access multiple reports (labels) for the same input feature, while our real-world datasets usually have only one noisy label for each feature.

2 Preliminaries

Instances Traditional classification problems with perfect supervisions build on a clean dataset $D := \{(x_n, y_n)\}_{n\in[N]}$, where $[N] := \{1, 2, \cdots, N\}$. Each *clean instance* (x_n, y_n) includes *feature* x_n and *clean label* y_n, which is drawn according to random variables $(X, Y) \sim \mathcal{D}$. In a more

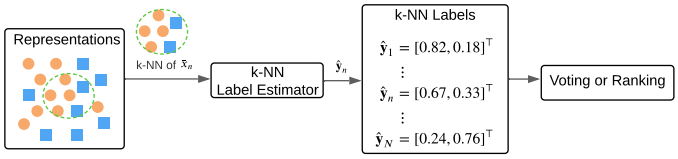

Figure 1: Detect noisy labels with similar representations. Orange circle: instance with noisy label 1. Blue square: instance with noisy label 2.

practical case, the clean labels are likely to be unavailable and the learner could only observe a noisy dataset denoted by $\widetilde{D} := \{(x_n, \tilde{y}_n)\}_{n\in[N]}$, where $(x_n, \tilde{y}_n)$ is a *noisy instance* and the *noisy label* $\tilde{y}_n$ may or may not be identical to y_n. We call $\tilde{y}_n$ is *corrupted* if $\tilde{y}_n \neq y_n$ and clean otherwise. The instance $(x_n, \tilde{y}_n)$ is a *corrupted instance* if $\tilde{y}_n$ is corrupted. The noisy data distribution corresponds to $\widetilde{D}$ is $(X, \widetilde{Y}) \sim \widetilde{\mathcal{D}}$. We focus on the closed-set label noise that Y and $\widetilde{Y}$ are assumed to be in the same label space, e.g., $Y, \widetilde{Y} \in [K]$ for a K-class classification task. Explorations on open-set data (Xia et al., 2020a; Luo et al., 2021) are deferred to future works.

Representations The *representation* of feature x_n is denoted by $\bar{x}_n := g(x_n)$, where $g(\cdot)$ denotes a representation extractor. Feature x_n could be in any shape and representation $\bar{x}_n$ is generally a high-dimensional vector. In this paper, we focus on a setting where the distances between two representations should be comparable or clusterable (Zhu et al., 2021b), i.e., nearby representations should belong to the same true class with a high probability (Gao et al., 2016), which could be defined as (k, δ_k) label clusterability in Definition 1.

Definition 1 ((k, δ_k) label clusterability)**.** *Given a representation extractor $g(\cdot)$, a dataset D satisfies (k, δ_k) label clusterability if: $\forall n \in [N]$, the representation $\bar{x}_n := g(x_n)$ and its k-Nearest-Neighbors (k-NN) $\bar{x}_{n_1}, \cdots, \bar{x}_{n_k}$ belong to the same true class with probability at least $1 - \delta_k$.*

Note δ_k captures two types of randomnesses: one comes from a probabilistic $Y|X$, i.e., $\exists i \in [K], \mathbb{P}(Y = i|X) \notin \{0, 1\}$; the other depends on the quality of representation and the value of k, which will be further illustrated in Figure 2. The $(k, 0)$ label clusterability is also known as k-NN label clusterability (Zhu et al., 2021b). It has been shown in (Zhu et al., 2021b) that k-NN label clusterability can often be satisfied for a reasonable representation when k is moderate to small. We define *k perfect extractors/representations* in Definition 2.

Definition 2 (k perfect extractor/representation)**.** *A representation extractor $g(\cdot)$ is called k perfect extractor if it induces k-NN label clusterability. Its outputs are called k perfect representations.*

Label noise detection Our paper aims to improve the performance of the label noise detection (a.k.a, detecting noisy/corrupted labels, finding label errors) which is measured by the F_1-score of the detected corrupted instances, which is the harmonic mean of the precision and recall, i.e. $F_1 = 2/(\texttt{Precision}^{-1} + \texttt{Recall}^{-1})$. Let $\mathbb{1}(\cdot)$ be the indicator function that takes value 1 when the specified condition is satisfied and 0 otherwise. Let $v_n = 1$ indicate that $\tilde{y}_n$ is detected as a corrupted label, and $v_n = 0$ if $\tilde{y}_n$ is detected to be clean. Then the precision and recall can be calculated as $\texttt{Precision} = \frac{\sum_{n\in[N]} \mathbb{1}(v_n=1, \tilde{y}_n \neq y_n)}{\sum_{n\in[N]} \mathbb{1}(v_n=1)}$, $\texttt{Recall} = \frac{\sum_{n\in[N]} \mathbb{1}(v_n=1, \tilde{y}_n \neq y_n)}{\sum_{n\in[N]} \mathbb{1}(\tilde{y}_n \neq y_n)}$.

3 Label Noise Detection Using Similar Representations

Different from most methods that detect noisy labels based on the logit layer or model predictions (Northcutt et al., 2021a; Cheng et al., 2021; Pruthi et al., 2020), we focus on representations of features. Particularly we are interested in the possibility of detecting noisy labels in a training-free way, once given good representations. In this section, we will first introduce intuitions, and then provide two efficient algorithms to detect label noise with similar representations.

3.1 Intuitions

The training-based detection methods often make decisions by comparing model predictions with noisy labels (Cheng et al., 2021; Northcutt et al., 2021a). However, the representation $g(x_n)$ cannot

be directly compared with the noisy label $\tilde{y}_n$ since $g(x_n)$ is not directly categorical, i.e., the output of a single $g(x_n)$ rarely corresponds to any label class. Thus the first step of our training-free solution should be establishing an auxiliary categorical information using only representations.

As illustrated in Figure 1, the high-level intuition is to check label consensuses of nearby representations. With k perfect representations as defined in Definition 2, we know the true labels of $\bar{x}_n$ and its k-NN $\bar{x}_{n_1}, \cdots, \bar{x}_{n_k}$ should be the same. If the label noise across these instances are group-dependent (Wang et al., 2021), we can treat their noisy labels as $k+1$ independent observations of $\mathbb{P}(\widetilde{Y}|X = x_n, Y = y_n)$, then estimate the probability by counting the frequency or weighted frequency[1] of each class in the *k-NN label estimator*, and get *k-NN labels* $\hat{\boldsymbol{y}}_n$. The i-th element $\hat{y}_n[i]$ can be interpreted as the estimated probability of predicting class-i.

Representations could be better than model predictions During supervised training, memorizing noisy labels makes the model generalizes poorly (Li et al., 2020a; Han et al., 2020), while using only representations may effectively avoid this issue. This is because representations often come from techniques beyond memorization such as unsupervised learning or self-supervised learning where the noisy labels are referred in representation learning. On the other hand, the extractor $g(\cdot)$ has great potential in domain adaption or transfer, which makes representations more flexible than the specific model predictions. We will demonstrate this point in Section 5.

3.2 Voting-Based Local Detection

Inspired by the idea implemented in model decisions, i.e., selecting the most likely class as the true class, we can simply "predict" the index that corresponds to the largest element in $\hat{\boldsymbol{y}}_n$ with random tie-breaking, i.e., $y_n^{\texttt{vote}} = \arg\max_{i \in [K]} \hat{y}_n[i]$. To further detect whether $\tilde{y}_n$ is corrupted or not, we only need to check $v_n := \mathbb{1}(y_n^{\texttt{vote}} \neq \tilde{y}_n)$. Recall $v_n = 1$ indicates a corrupted label. This voting method relies only on the local information within each k-NN label $\hat{\boldsymbol{y}}_n$, which may not be robust with imperfect representations. Intuitively, when the gap between the true class probability and the wrong class probability is small, the majority vote will be likely to make mistakes due to sampling errors in $\hat{\boldsymbol{y}}_n$. Thus only using local information within each $\hat{\boldsymbol{y}}_n$ may not be sufficient. It is important to leverage more information such as some global statistics, which will be discussed later.

3.3 Ranking-Based Global Detection

From a global perspective, if the likelihood for each instance being clean could be evaluated by some scoring functions, we can sort the scores in an increasing order and filter out the low-score instances as corrupted ones. Based on this intuition, there are two critical components: the *scoring function* and the *threshold* to differentiate the low-score part (corrupted) and the high-score part (clean).

Scoring function A good scoring function should be able to give clean instances higher scores than corrupted instances. We adopt cosine similarity defined as: $\mathsf{Score}(\hat{\boldsymbol{y}}_n, j) = \frac{\hat{\boldsymbol{y}}_n^\top \boldsymbol{e}_j}{\|\hat{\boldsymbol{y}}_n\|_2 \|\boldsymbol{e}_j\|_2}$, where $\boldsymbol{e}_j$ is the one-hot encoding of label j. To evaluate whether the soft label $\hat{\boldsymbol{y}}_n$ informs us a clean instance or not, we compare $\mathsf{Score}(\hat{\boldsymbol{y}}_n, \tilde{y}_n)$ with other instances that have the same noisy label. This scoring function captures more information than majority votes, which is summarized as follows.

Property 1 (Relative score)**.** *With in the same instance, the score of the majority class is higher than the others, i.e.* $\mathsf{Score}(\hat{\boldsymbol{y}}_n, y_n^{\texttt{vote}}) > \mathsf{Score}(\hat{\boldsymbol{y}}_n, j), \forall j \neq y_n^{\texttt{vote}}, j \in [K], \forall n \in [N]$.

Property 2 (Absolute score)**.** $\mathsf{Score}(\hat{\boldsymbol{y}}_n, j)$ *is jointly determined by both* $\hat{y}_n[j]$ *and* $\hat{y}_n[j'], \forall j' \neq j$.

The first property guarantees that the corrupted labels would have lower scores than clean labels for the same instance when the vote is correct. However, although solely relying on Property 1 may work well in the voting-based method which makes decisions individually for each instance, it is not sufficient to be trustworthy in the ranking-based global detection. The main reason is that, across different instances, the non-majority classes of some instances may have higher absolute scores than the majority classes of the other instances, which is especially true for general instance-dependent label noise with heterogeneous noise rates (Cheng et al., 2021). Property 2 helps make it less likely to happen. Consider an example as follows.

[1] One can weight instances by the similarity to the center instance.

Example Suppose $\hat{\boldsymbol{y}}_{n_1} = \hat{\boldsymbol{y}}_{n_2} = [0.6, 0.4, 0.0]^\top$, $\hat{\boldsymbol{y}}_{n_3} = [0.34, 0.0.33, 0.33]^\top$, $y_{n_1} = y_{n_2} = y_{n_3} = 1$, $\tilde{y}_{n_1} = \tilde{y}_{n_3} = 1, \tilde{y}_{n_2} = 2$. We can use the majority vote to get perfect detection in this case, i.e., $y_{n_1}^{\texttt{vote}} = y_{n_2}^{\texttt{vote}} = y_{n_3}^{\texttt{vote}} = 1 = y_{n_1}$, since the first class of each instance has the largest value. However, if we directly use a single value in soft label $\boldsymbol{y}_n$ to score them, e.g., $\mathsf{Score}^{\prime}(\hat{\boldsymbol{y}}_n, j) = \hat{y}_n[j]$, we will have $\hat{y}_{n_1}[\tilde{y}_{n_1}] = 0.6 > \hat{y}_{n_2}[\tilde{y}_{n_2}] = 0.4 > \hat{y}_{n_3}[\tilde{y}_{n_3}] = 0.33$, where the ranking is $n_3 \prec n_2 \prec n_1$. Ideally, we know instance n_2 is corrupted and the true ranking should be $n_2 \prec n_3 \prec n_1$ or $n_2 \prec n_1 \prec n_3$. To mitigate this problem, we choose the cosine similarity as our scoring function. The three instances could be scored as $0.83, 0.55, 0.59$, corresponding to an ideal ranking $n_2 \prec n_3 \prec n_1$. We formally introduce the detailed ranking approach as follows.

Ranking Suppose we have a group of instances with the same noisy class j, i.e. $\{(x_n, \tilde{y}_n)\}_{n \in \mathcal{N}_j}$, where $\mathcal{N}_j := \{n | \tilde{y}_n = j\}$ are the set of indices that correspond to noisy class j. Let N_j be the number of indices in $\mathcal{N}_j$ (counted from noisy labels). Intuitively, we can first sort all instances in $\mathcal{N}_j$ in an increasing order by `argsort` and obtain the original indices for the sorted scores as: $\mathcal{I} = \texttt{argsort}\{\mathsf{Score}(\hat{\boldsymbol{y}}_n, j)\}_{n \in \mathcal{N}_j}$, where the low-score head is supposed to consist of corrupted instances (Northcutt et al., 2021a). Then we can simply select the first $\widetilde{N}_j$ instances with low scores as corrupted instances: $v_n = \mathbb{1}(\texttt{Loc}(n, \mathcal{I}) \leq \widetilde{N}_j)$, where $\texttt{Loc}(n, \mathcal{I})$ returns the index of n in $\mathcal{I}$. Instead of manually tuning $\widetilde{N}_j$, we discuss how to determine it algorithmically.

Threshold The number of corrupted instances in $\mathcal{N}_j$ is approximately $\mathbb{P}(Y \neq j | \widetilde{Y} = j) \cdot N_j$ when N_j is sufficiently large. Therefore if all the corrupted instances have lower scores than any clean instance, we can set $\widetilde{N}_j = \mathbb{P}(Y \neq j | \widetilde{Y} = j) \cdot N_j$ to obtain the ideal division. Note N_j can be obtained by directing counting the number of instances with noisy label j. To calculate the probability $\mathbb{P}(Y \neq j | \widetilde{Y} = j) = 1 - \mathbb{P}(Y = j | \widetilde{Y} = j)$, we borrow the results from the HOC estimator (Zhu et al., 2021b), where the noise transition probability $\mathbb{P}(\widetilde{Y} = j | Y = j)$ and the marginal distribution of clean label $\mathbb{P}(Y = j)$ can be estimated with only representations and the corresponding noisy labels. Then we can calculated our needed probability by Bayes' rule $\mathbb{P}(Y = j | \widetilde{Y} = j) = \mathbb{P}(\widetilde{Y} = j | Y = j) \cdot \mathbb{P}(Y = j) / \mathbb{P}(\widetilde{Y} = j)$, where $\mathbb{P}(\widetilde{Y} = j)$ can be estimated by counting the frequency of noisy label j in $\widetilde{D}$. Technically other methods exist in the literature to estimate $\mathbb{P}(\widetilde{Y} | Y)$ (Liu & Tao, 2015; Patrini et al., 2017; Northcutt et al., 2021a; Li et al., 2021). But they often require training a model to fit the data distribution, which conflict with our goal of a training-free solution; instead, HOC fits us perfectly.

3.4 Algorithm: Detection with Similar Representations (SimiRep)

Algorithm 1 summarizes our solution. The main computation complexity is obtaining representations with extractor $g(\cdot)$, which is less than the cost of evaluating the model compared with the training-based methods. Thus SimiRep can filter out corrupted instances efficiently. In Algorithm 1, we run either voting-based local detection as Lines 7, 8, or ranking-based global detection as Lines 14, 15. The detection is run multiple times with random standard data augmentations to reduce the variance of estimation. The majority of results from different epochs is adopted as the final detection output as Line 20, i.e., flag as corrupted if $v_n = 1$ in more than half of the epochs.

4 How Do Representations Affect Our Solution?

In this section, we will first show how a good but imperfect representation[2] affects the selection of the hyperparameter k. We then offer an analysis of the error upper bound for the ranking-based method given a representation extractor $g(\cdot)$.

4.1 How Do Representations Affect the Choice of k?

Recall k is used as illustrated in Figure 1. On one hand, the k-NN label estimator will be more accurate if there is stronger clusterability that more neighbor representations belong to the same true class (Liu & Liu, 2015; Zhu et al., 2021b), which helps improve the performance of later algorithms. On the other hand, with good but imperfect representations, stronger clusterability with a larger k

[2]Note the voting-based method achieves an F_1-score of 1 given k perfect representations, $k \to +\infty$.

Algorithm 1 Detection with **Simi**lar **Rep**resentations (The SimiRep Detector)

Input: Number of epochs: M. k-NN parameter: k. Noisy dataset: $\widetilde{D} = \{(x_n, \tilde{y}_n)\}_{n\in[N]}$. Representation extractor: $g(\cdot)$. Method: *Vote* or *Rank*. Epoch counter $m = 0$.
repeat
  $x'_n \leftarrow \texttt{RandPreProcess}(x_n), \forall n$; *# Initialize & Standard data augmentations*
  $\bar{x}_n \leftarrow g(x'_n), \forall n$; *# Extract representations with $g(\cdot)$*
  $\hat{\boldsymbol{y}}_n \leftarrow \texttt{kNNLabel}(\{\bar{x}_n\}_{n\in[N]}, k)$ *# Get soft labels based on the clusterability of representations*
  if *Vote* **then**
    $y_n^{\texttt{vote}} \leftarrow \arg\max_{i\in[K]} \hat{y}_n[i]$; *# Apply local majority vote*
    $v_n \leftarrow \mathbb{1}(y_n^{\texttt{vote}} \neq \tilde{y}_n), \forall n \in [N]$; *# Treat as corrupted if majority votes agree with noisy labels*
  else
    $\mathbb{P}(Y), \mathbb{P}(\widetilde{Y}|Y) \leftarrow \texttt{HOC}(\{(\bar{x}_n, \tilde{y}_n)\}_{n\in[N]})$;
    # Estimate clean priors $\mathbb{P}(Y)$ and noise transitions $\mathbb{P}(\widetilde{Y}|Y)$ by the HOC estimator
    $\mathbb{P}(Y|\widetilde{Y}) = \mathbb{P}(\widetilde{Y}|Y) \cdot \mathbb{P}(Y)/\mathbb{P}(\widetilde{Y})$; *# Estimate thresholds by Bayes' rule*
    for j **in** $[K]$ **do**
      $\mathcal{N}_j := \{n|\tilde{y}_n = j\}$; *# Detect corrupted labels in each set $\mathcal{N}_j$*
      $\mathcal{I} \leftarrow \texttt{argsort}\{\mathsf{Score}(\hat{\boldsymbol{y}}_n, j)\}_{n\in\mathcal{N}_j}$; *# $\mathcal{I}$ records the raw index of each sorted value*
      $v_n \leftarrow \mathbb{1}\big(\texttt{Loc}(n, \mathcal{I}) \leq \lfloor(1 - \mathbb{P}(Y = j|\widetilde{Y} = j)) \cdot N_j\rfloor\big)$;
      # Select low-score (head) instances as corrupted ones
    end for
  end if
  $\mathcal{V}_m = \{v_n\}_{n\in[N]}$; *# Record detection results in the m-th epoch*
until M times
$\mathcal{V} = \texttt{Vote}(\mathcal{V}_m, \forall m \in [M])$; *# Do majority vote based on results from M epochs*
Output: $[N] \setminus \mathcal{V}$.

is less likely to satisfy, thus the violation probability δ_k increases with k for a given extractor $g(\cdot)$. We take the voting-based method as an example and analyze this tradeoff. For a clean presentation, we focus on a binary classification with instance-dependent label noise where $\mathbb{P}(Y = 1) = p$, $\mathbb{P}(\widetilde{Y} = 2|X, Y = 1) = e_1(X)$, $\mathbb{P}(\widetilde{Y} = 1|X, Y = 2) = e_2(X)$. Suppose the instance-dependent noise rate is upper-bounded by e, i.e., $e_1(X) \leq e, e_2(X) \leq e$. With δ_k as in Definition 1, we calculate the lower bound of the probability that the vote is correct in Proposition 1.

Proposition 1. *The lower bound for the probability of getting true detection with majority vote is*

$$\mathbb{P}(\textit{Vote is correct}|k) \geq (1 - \delta_k) \cdot I_{1-e_1}(k + 1 - k', k' + 1),$$

where $k' = \lceil (k + 1)/2 \rceil - 1$, $I_{1-e_1}(k + 1 - k', k' + 1)$ is the regularized incomplete beta function defined as $I_{1-e}(k + 1 - k', k' + 1) = (k + 1 - k')\binom{k+1}{k'} \int_0^{1-e} t^{k-k'}(1 - t)^{k'} dt$.

Proposition 1 shows the tradeoff between a reliable k-NN label and an accurate vote. When k is increasing, **Term-1** $(1 - \delta_k)$ (quality of representations) decreases but **Term-**2 $I_{1-e_1}(k + 1 - k', k' + 1)$ (result of pure majority vote) increases. With Proposition 1, we are ready to answer the question: *when do we need more labels?* See Remark 1.

Remark 1. *Consider the lower bounds with k_1 and k_2 ($k_1 < k_2$). Supposing the first lower bound is lower than the second lower bound, based on Proposition 1, we roughly study the trend with an increasing k by comparing two bounds and get*

$$\frac{1 - \delta_{k_1}}{1 - \delta_{k_2}} < \frac{I_{1-e}(k_2 + 1 - k'_2, k'_2 + 1)}{I_{1-e}(k_1 + 1 - k'_1, k'_1 + 1)}.$$

For example, when $k_1 = 5$, $k_2 = 20$, $e = 0.4$, we can calculate the incomplete beta function and $\frac{1-\delta_5}{1-\delta_{20}} < 1.52$. Supposing $\delta_5 = 0.2$, we have $\delta_{20} < 0.47$. This indicates increasing k from 5 to 20 would not improve the lower bound with imperfect representations with $\delta_{20} > 0.47$. This observation helps us set k with a practical and imperfect $g(\cdot)$. We set $k = 10$ in all of our experiments.

Remark 1 indicates that: ***with imperfect representations, a small k may achieve the best (highest) probability lower bound***. To further consolidate this claim, we numerically calculate δ_k with different $g(\cdot)$ and the corresponding probability lower bound in Figure 2. We find most of the probability

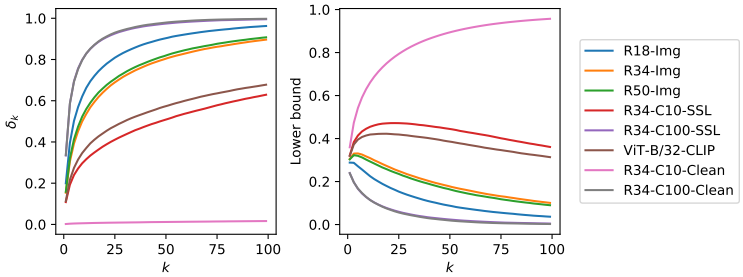

Figure 2: The trends of δ_k and probability lower bounds on CIFAR-10 (Krizhevsky et al., 2009) with different $g(\cdot)$. The outputs of the last convolution layer are adopted. R18/34/50: ResNet18/34/50. Img: Pre-trained on ImageNet (Deng et al., 2009). C10/100-Clean: Pre-trained on clean CIFAR-10/100. C10/100-SSL: Pre-trained on CIFAR-10/100 without labels by SimCLR (Chen et al., 2020). ViT-B/32-CLIP: CLIP Pre-trained vision transformer (Radford et al., 2021).

lower bounds first increase then decrease except for the perfect $g(\cdot)$ which is trained using ground-truth labels. Note the perfect $g(\cdot)$ has memorized all clean instances so that $\delta_k \to 0$ since $k \ll 5000$ (the number of instances in the same label class).

4.2 How Do Representations Affect F_1-Score?

We next prove the probability bound for the performance of the ranking-based method. Consider a K-class classification problem with informative instance-dependent label noise (Cheng et al., 2021). Denote random variable S by the score of each instance being clean. A higher score S indicates the instance is more likely to be clean. Then for instances in $\mathcal{N}_j$, we have two set of random variables $\mathcal{S}_j^{\text{true}} := \{S^{\text{true}} | n \in \mathcal{N}_j, \tilde{y}_n = y_n\}$ and $\mathcal{S}_j^{\text{false}} := \{S^{\text{false}} | n \in \mathcal{N}_j, \tilde{y}_n \neq y_n\}$. Intuitively, the score S_j^{true} should be greater than S_j^{false}. Suppose their means, which depend on noise rates, are bounded, i.e., $\mathbb{E}[S_j^{\text{true}}] \geq \mu_j^{\text{true}}$, $\mathbb{E}[S_j^{\text{false}}] \leq \mu_j^{\text{false}}$. Assume there exists a feasible v such that both S_j^{true} and S_j^{false} follow sub-Gaussian distributions with variance proxy $\frac{\Delta^2}{2v}$ (Buldygin & Kozachenko, 1980; Zhu et al., 2021c) such that: $\mathbb{P}(\mu_j^{\text{true}} - S_j^{\text{true}} \geq t) \leq e^{-v(t/\Delta)^2}$, and $\mathbb{P}(S_j^{\text{false}} - \mu_j^{\text{false}} \geq t) \leq e^{-v(t/\Delta)^2}$, where $1/\Delta$ is the "height" of both distributions, i.e., $\mathbb{P}(S_j^{\text{true}} = \mu_j^{\text{true}}) = \mathbb{P}(S_j^{\text{false}} = \mu_j^{\text{false}}) = 1/\Delta$, v is the decay rate of tails. Let N_j^- (N_j^+) be the number of indices in $\mathcal{S}_j^{\text{false}}$ ($\mathcal{S}_j^{\text{true}}$). Theorem 1 summarizes the performance bound of the ranking-based method. See Appendix for the proof.

Theorem 1. *With probability at least p, the F_1-score of detecting corrupted instances in $\mathcal{N}_j$ with the rank-based method by threshold $N_j \mathbb{P}(Y = j | \widetilde{Y} = j)$ is at least $1 - \frac{e^{-v} \max(N^-, N^+) + \alpha}{N^-}$, where $p = \int_{-1}^{\mu^{true} - \mu^{false} - \Delta} f(t)dt$, $f(t)$ is the probability density function of the difference of two independent beta-distributed random variables $\beta_1 - \beta_2$, where $\beta_1 \sim Beta(N^-, 1), \beta_2 \sim Beta(\alpha + 1, N^+ - \alpha)$.*

Theorem 1 shows the performance of detection depends on 1) the concentration of S_j^{true} and S_j^{false} (denoted by variance proxy $\frac{\Delta^2}{2v}$); 2) the distance between S_j^{true} and S_j^{false} (denoted by $\mu^{\text{true}} - \mu^{\text{false}}$). Intuitively, with proper scoring function and good representations, we have small variance proxy (small Δ and large v) and F_1-score approximates to 1.

5 Empirical Results

We present experimental evidence in this section. The performance is measured by the F_1-score of the detected corrupted labels as defined in Section 2. Note there is no any training procedure in our method. The only hyperparameters in our methods are the number of epochs M and the k-NN parameter k. Intuitively, a larger M returns a collective result from more times of detection, which should be more accurate. The hyperparameter k cannot be set too large as demonstrated in Figure 2. In our experiments, rather than fine-tune M and k for different settings, we fix $M = 21$ and $k = 10$.

Table 1: Comparisons of F_1-scores (%). CORES, CL, TracIn: Train with noisy supervisions. SimiRep-V and SimiRep-R: Get $g(\cdot)$ without any supervision. Top 2 are **bold**.

Method	*CIFAR10*				*CIFAR100*			
	Human	*Symm. 0.6*	*Asym. 0.3*	*Inst. 0.4*	*Human*	*Symm. 0.6*	*Asym. 0.3*	*Inst. 0.4*
CORES	65.00	92.94	7.68	**87.43**	3.52	**92.34**	0.02	9.67
CL	55.85	80.59	76.45	62.89	64.58	78.98	52.96	50.08
TracIn	55.02	76.94	73.47	58.85	61.75	76.74	48.42	49.89
SimiRep-V	**82.30**	**93.21**	**82.52**	81.09	**73.19**	84.48	**65.42**	**74.26**
SimiRep-R	**83.28**	**95.56**	**83.58**	**82.26**	**74.67**	**88.68**	**62.89**	**73.53**

Synthetic label noise We experiment with three popular synthetic label noise models: the *symmetric* label noise, the *asymmetric* label noise, and the *instance-dependent* label noise. Denote the ratio of instances with corrupted labels in the whole dataset by η. Both the symmetric and the asymmetric noise models follow the class-dependent assumption (Liu & Tao, 2015), i.e., the label noise only depends only on the clean class: $\mathbb{P}(\widetilde{Y}|X,Y) = \mathbb{P}(\widetilde{Y}|Y)$. Specially, the symmetric noise is generated by uniform flipping, i.e., randomly flipping a true label to the other possible classes w.p. η (Cheng et al., 2021). The asymmetric noise is generated by pair-wise flipping, i.e., randomly flipping true label i to the next class $(i \mod K) + 1$. Denote by d the dimension of features. The instance-dependent label noise is synthesized by randomly generating a $d \times K$ projection matrix w_i for each class i and project each incoming feature with true class y_n onto each column of w_{y_n} (Xia et al., 2020b). Instance n is more likely to be flipped to class j if the projection value of x_n on the j-th column of w_{y_n} is high. See Appendix B in Xia et al. (2020b) and Appendix D.1 in Zhu et al. (2021b) for more details. We use symmetric noise with $\eta = 0.6$ (*Symm. 0.6*), asymmetric noise with $\eta = 0.3$ (*Asym. 0.3*), and instance-dependent noise with $\eta = 0.4$ (*Inst. 0.4*) in experiments.

Real-world label noise The *real-world* label noise comes from human annotations. We use the $50,000$ noisy training labels ($\eta \approx 0.16$) for CIFAR-10 collected by Zhu et al. (2021b), and our self-collected $50,000$ noisy training labels ($\eta \approx 0.40$) for CIFAR-100 (data will be released in the non-anonymous version). Both sets of noisy labels are crowd-sourced from Amazon Mechanical Turk. Our solution can be easily applied on large-scale datasets such as Clothing1M (Xiao et al., 2015) by using common pre-trained models. But it is hard to verify the performance since the ground-truth labels are not available.

5.1 Supervision May Not Be Necessary

Our first experiment aims to show that training with noisy supervisions may not be necessary in detecting corrupted labels. To this end, we compare our methods, i.e., voting-based local detection (SimiRep-V) and ranking-based global detection (SimiRep-R), with three recent noise detection works: CORES (Cheng et al., 2021), confident learning (CL) (Northcutt et al., 2021a), and TracIn (Pruthi et al., 2020). We use ResNet34 as the backbone network in this experiment.

Baseline settings All these three baselines require training a model with the noisy supervision. Specifically, CORES (Cheng et al., 2021) trains ResNet34 on the noisy dataset and uses its proposed sample sieve to filter out the corrupted instances. We adopt its default setting during training and calculate the F_1-score of the sieved out corrupted instances. Confident learning (CL) (Northcutt et al., 2021a) detects corrupted labels by firstly estimating probabilistic thresholds to characterize label noise, ranking instances based on model predictions, then filtering out corrupted instances based on ranking and thresholds. We adopt its default hyper-parameter setting to train ResNet34. TracIn (Pruthi et al., 2020) detects corrupted labels by evaluating the self-influence of each instance, where the corrupted instances tend to have a high influence score. The influence scores are calculated based on gradients of the last layer of ResNet34 at epoch $40, 50, 60, 100$, where the model is trained with a batch size of 128. The initial learning rate is 0.1 and decays to 0.01 at epoch 50. Note TracIn only provides ranking for instances. To exactly detect corrupted instances, thresholds are required. For a fair comparison, we refer to the thresholds learned by confident learning (Northcutt et al., 2021a). Thus the corrupted instances selected by TracIn are based on the ranking from its self-influence and thresholds from CL. To highlight that our solutions work well without any supervision, our representation extractor $g(\cdot)$ comes from the ResNet34 pre-trained by SimCLR (Chen et al., 2020) where contrastive learning is applied and *no supervision* is required. Extractor $g(\cdot)$ is obtained with only in-distribution features, e.g., for experiments with CIFAR-10, $g(\cdot)$ is pre-trained with features only from CIFAR-10.

Table 2: Comparisons of F_1-scores (%). CORES, CL, TracIn: Use logit layers. SimiRep-V and SimiRep-R: Use only representations. Top 2 are **bold**.

Method	*CIFAR10*				*CIFAR100*			
	Human	*Symm. 0.6*	*Asym. 0.3*	*Inst. 0.4*	*Human*	*Symm. 0.6*	*Asym. 0.3*	*Inst. 0.4*
CE Sieve	67.21	94.56	5.24	8.41	16.24	88.55	2.6	1.63
CORES	83.18	**96.94**	12.05	**88.89**	38.52	**92.33**	7.02	**85.52**
CL	69.76	95.03	77.14	62.91	67.64	85.67	62.58	61.53
TracIn	81.85	95.96	80.75	64.97	**79.32**	**91.03**	63.12	64.31
SimiRep-V	**87.43**	96.44	**88.97**	87.11	76.26	86.88	**73.50**	**80.03**
SimiRep-R	**87.45**	**96.74**	**89.04**	**91.14**	**79.21**	90.54	**68.14**	77.37

Performance Table 1 compares the results obtained with or without supervisions. We can see both the voting-based and the ranking-based method achieve overall higher F_1-scores compared with the other three results that require learning with noisy supervisions. Moreover, in detecting the real-world human-level noisy labels, our solution outperforms baselines around 20% on CIFAR-10 and 10% on CIFAR-100, which indicates the training-free solution are more robust to complicated noise patterns. One might also note that CORES achieves exceptionally low F_1-scores on CIFAR-10/100 with asymmetric noise and CIFAR-100 with human noise. This observation also informs us that customized training processes might not always be universally applicable.

5.2 Representation Layers May Perform Better Than Logit Layers

Table 2 shows the detection using either representation layers or logit layers of DNNs. In addition to the baselines compared in Section 5.1, we also compare to CE Sieve (Cheng et al., 2021) which follows the same sieving process as CORES but uses CE loss without regularizer. All methods adopt ViT-B/32 pre-trained by CLIP (Radford et al., 2021). Specifically, the training-based methods further fine-tune a linear logit layer on noisy supervisions, while our solution directly sets this pre-trained model as $g(\cdot)$. Other settings are the same as those in Section 5.1. By counting the frequency of reaching top-2 F_1-scores, we find SimiRep-R wins 1st place, SimiRep-V and CORES are tied for 2nd place. However, similar to Table 2, we find the training process of CORES to be unstable. For instance, it almost fails for CIFAR-100 with asymmetric noise. It is therefore reasonable to believe both methods with only representations achieve an overall higher F_1-score than other methods with logit layers. In other words, compared with our methods, the extra fine-tuning of logit layers with noisy supervisions cannot always help improve the performance of detecting corrupted labels.

5.3 What Are Good Representations

Table 3: Comparisons of F_1-scores (%) using $g(\cdot)$ with different δ_k (%). Model names are the same as Figure 2.

Pre-trained Model	*CIFAR10*			*CIFAR100*		
	$1-\delta_k$	*Human*	*Inst. 0.4*	$1-\delta_k$	*Human*	*Inst. 0.4*
R18-Img	35.73	75.40	80.22	11.30	74.91	71.99
R34-Img	48.13	79.52	82.43	16.17	76.88	74.00
R50-Img	45.77	78.40	82.06	15.81	76.55	73.51
ViT-B/32-CLIP	64.12	87.45	91.14	19.94	79.21	77.37
R34-C10-SSL	69.31	83.28	85.26	2.59	68.03	65.94
R34-C10-Clean	99.41	98.39	98.59	0.22	60.90	60.73
R34-C100-SSL	18.59	59.96	74.99	22.46	74.67	73.53
R34-C100-Clean	18.58	60.17	76.41	89.07	92.87	95.29

Previous experiments demonstrate our methods overall outperform baselines given representations from contrastive pre-training. It is interesting to see how other representations perform. We summarize results of SimiRep-R in Table 3. There are several interesting findings: 1) The ImageNet pre-trained models perform well, indicating *the traditional supervised training on out-of-distribution data helps*; 2) For CIFAR-100, extractor $g(\cdot)$ obtained with only features from CIFAR-10 (R34-C10-SSL) performs better than the extractor with clean CIFAR-10 (R34-C10-Clean), indicating that *contrastive pre-training has better generalization ability to out-of-distribution data than supervised learning*; 3) The F_1-scores achieved by $g(\cdot)$ trained with the corresponding clean dataset are close to 1, indicating *our solution can give perfect detection with ideal representations*.

6 Conclusions

This paper proposed a new and universally applicable training-free solution to detect noisy labels by using the neighborhood information defined by a good set of representations. We have also demonstrated that good representations are reasonable assumptions with pre-training or self-supervised learning. Future works will explore other tasks that could benefit from label cleaning.

REFERENCES

Vibhu Agarwal, Tanya Podchiyska, Juan M Banda, Veena Goel, Tiffany I Leung, Evan P Minty, Timothy E Sweeney, Elsie Gyang, and Nigam H Shah. Learning statistical models of phenotypes using noisy labeled training data. *Journal of the American Medical Informatics Association*, 23 (6):1166–1173, 2016.

Ehsan Amid, Manfred KK Warmuth, Rohan Anil, and Tomer Koren. Robust bi-tempered logistic loss based on bregman divergences. In *Advances in Neural Information Processing Systems*, pp. 14987–14996, 2019.

Dara Bahri, Heinrich Jiang, and Maya Gupta. Deep k-nn for noisy labels. In *International Conference on Machine Learning*, pp. 540–550. PMLR, 2020.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*, 2019.

Valerii V Buldygin and Yu V Kozachenko. Sub-gaussian random variables. *Ukrainian Mathematical Journal*, 32(6):483–489, 1980.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=2VXyy9mIyU3`.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Wei Gao, Bin-Bin Yang, and Zhi-Hua Zhou. On the resistance of nearest neighbor to random noisy labels. *arXiv preprint arXiv:1607.07526*, 2016.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Maoguo Gong, Hao Li, Deyu Meng, Qiguang Miao, and Jia Liu. Decomposition-based evolutionary multiobjective optimization to self-paced learning. *IEEE Transactions on Evolutionary Computation*, 23(2):288–302, 2018.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pp. 8527–8537, 2018.

Bo Han, Quanming Yao, Tongliang Liu, Gang Niu, Ivor W Tsang, James T Kwok, and Masashi Sugiyama. A survey of label-noise representation learning: Past, present and future. *arXiv preprint arXiv:2011.04406*, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021.

Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9865–9874, 2019.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304–2313. PMLR, 2018.

Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning*, pp. 4804–4815. PMLR, 2020.

David R Karger, Sewoong Oh, and Devavrat Shah. Efficient crowdsourcing for multi-class labeling. In *Proceedings of the ACM SIGMETRICS/international conference on Measurement and modeling of computer systems*, pp. 81–92, 2013.

D.R. Karger, S. Oh, and D. Shah. Iterative learning for reliable crowdsourcing systems. In *Neural Information Processing Systems*, NIPS '11, 2011.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Jingling Li, Mozhi Zhang, Keyulu Xu, John P Dickerson, and Jimmy Ba. Noisy labels can induce good representations. *arXiv preprint arXiv:2012.12896*, 2020a.

Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020b. URL `https://openreview.net/forum?id=HJgExaVtwr`.

Xuefeng Li, Tongliang Liu, Bo Han, Gang Niu, and Masashi Sugiyama. Provably end-to-end label-noise learning without anchor points. *arXiv preprint arXiv:2102.02400*, 2021.

Andy T Liu, Shang-Wen Li, and Hung-yi Lee. Tera: Self-supervised learning of transformer encoder representation for speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2351–2366, 2021.

Qiang Liu, Jian Peng, and Alexander Ihler. Variational inference for crowdsourcing. In *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 1*, pp. 692–700, 2012.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *arXiv preprint arXiv:2007.00151*, 2020.

Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.

Yang Liu. Understanding instance-level label noise: Disparate impacts and treatments. In *International Conference on Machine Learning*, pp. 6725–6735. PMLR, 2021.

Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20, 2020.

Yang Liu and Mingyan Liu. An online learning approach to improving the quality of crowd-sourcing. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):217–230, 2015.

Huixiang Luo, Hao Cheng, Yuting Gao, Ke Li, Mengdan Zhang, Fanxu Meng, Xiaowei Guo, Feiyue Huang, and Xing Sun. On the consistency training for open-set semi-supervised learning, 2021.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in neural information processing systems*, pp. 1196–1204, 2013.

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. *arXiv preprint arXiv:1910.01842*, 2019.

Curtis Northcutt, Lu Jiang, and Isaac Chuang. Confident learning: Estimating uncertainty in dataset labels. *Journal of Artificial Intelligence Research*, 70:1373–1411, 2021a.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. Pervasive label errors in test sets destabilize machine learning benchmarks. *arXiv preprint arXiv:2103.14749*, 2021b.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1944–1952, 2017.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19920–19930, 2020.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

Jun Shu, Qian Zhao, Keyu Chen, Zongben Xu, and Deyu Meng. Learning adaptive loss for robust learning with noisy labels. *arXiv preprint arXiv:2002.06482*, 2020.

Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pp. 270–279. Springer, 2018.

Jialu Wang, Yang Liu, and Caleb Levy. Fair classification with group-dependent label noise. FAccT, pp. 526–536, New York, NY, USA, 2021.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 322–330, 2019.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13726–13735, 2020.

Jiaheng Wei and Yang Liu. When optimizing f-divergence is robust with label noise. In *International Conference on Learning Representations*, 2021.

Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In *Advances in Neural Information Processing Systems*, pp. 6838–6849, 2019.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Jiankang Deng, Jiatong Li, and Yinian Mao. Extended T: Learning with mixed closed-set and open-set noisy labels. *arXiv preprint arXiv:2012.00932*, 2020a.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7597–7610, 2020b.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*, 2021.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2691–2699, 2015.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*, 2019.

Quanming Yao, Hansi Yang, Bo Han, Gang Niu, and James T Kwok. Searching to exploit memorization effect in learning with noisy labels. In *Proceedings of the 37th International Conference on Machine Learning*, ICML '20, 2020.

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 7164–7173. PMLR, 09–15 Jun 2019.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Jing Zhang, Victor S Sheng, Tao Li, and Xindong Wu. Improving crowdsourced label quality using noise correction. *IEEE transactions on neural networks and learning systems*, 29(5):1675–1688, 2017.

Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature-dependent label noise: A progressive approach. *arXiv preprint arXiv:2103.07756*, 2021.

Yuchen Zhang, Xi Chen, Dengyong Zhou, and Michael I Jordan. Spectral methods meet em: A provably optimal algorithm for crowdsourcing. *Advances in neural information processing systems*, 27:1260–1268, 2014.

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pp. 8778–8788, 2018.

Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with instance-dependent label noise. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021a.

Zhaowei Zhu, Yiwen Song, and Yang Liu. Clusterability as an alternative to anchor points when learning with noisy labels. In *Proceedings of the 38th International Conference on Machine Learning*, ICML '21, 2021b.

Zhaowei Zhu, Jingxuan Zhu, Ji Liu, and Yang Liu. Federated bandit: A gossiping approach. In *Abstract Proceedings of the 2021 ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*, pp. 3–4, 2021c.

The omitted proofs are provided as follows.

A THEORETICAL ANALYSES

A.1 PROOF FOR PROPOSITION 1

Now we derive a lower bound for the probability of getting true detection with majority vote:

$$\begin{aligned}\mathbb{P}(\text{Vote is correct}|k) \geq &(1-\delta_k)\cdot\Bigg[p\sum_{l=0}^{\lceil (k+1)/2\rceil -1}\binom{k+1}{l}e_1^l(1-e_1)^{k+1-l}\\ &+(1-p)\sum_{l=0}^{\lceil (k+1)/2\rceil -1}\binom{k+1}{l}e_2^l(1-e_2)^{k+1-l}\Bigg]\\ =&(1-\delta_k)\cdot[p\cdot I_{1-e_1}(k+1-k',k'+1)+(1-p)\cdot I_{1-e_2}(k+1-k',k'+1)]\end{aligned}$$

where $I_{1-e_1}(k+1-k',k'+1)$ is the regularized incomplete beta function defined as

$$I_{1-e}(k+1-k',k'+1)=(k+1-k')\binom{k+1}{k'}\int_0^{1-e}t^{k-k'}(1-t)^{k'}dt,$$

and $k'=\lceil (k+1)/2\rceil -1$.

B PROOF FOR THEOREM 1

Proof. Now we derive the worst-case error bound. We first repeat the notations defined in Section 4.2 as follows.

Denote random variable S by the score of each instance being clean. A higher score S indicates the instance is more likely to be clean. Then for instances in $\mathcal{N}_j$, we have two set of random variables $\mathcal{S}_j^{\text{true}} := \{S^{\text{true}}|n\in\mathcal{N}_j,\tilde{y}_n=y_n\}$ and $\mathcal{S}_j^{\text{false}} := \{S^{\text{false}}|n\in\mathcal{N}_j,\tilde{y}_n\neq y_n\}$. Intuitively, the score S_j^{true} should be greater than S_j^{false}. Suppose their means, which depend on noise rates, are bounded, i.e., $\mathbb{E}[S_j^{\text{true}}]\geq\mu_j^{\text{true}}$, $\mathbb{E}[S_j^{\text{false}}]\leq\mu_j^{\text{false}}$. Assume there exists a feasible v such that both S_j^{true} and S_j^{false} follow sub-Gaussian distributions with variance proxy $\frac{\Delta^2}{2v}$ (Buldygin & Kozachenko, 1980; Zhu et al., 2021c) such that: $\mathbb{P}(\mu_j^{\text{true}}-S_j^{\text{true}}\geq t)\leq e^{-v(t/\Delta)^2}$, and $\mathbb{P}(S_j^{\text{false}}-\mu_j^{\text{false}}\geq t)\leq e^{-v(t/\Delta)^2}$, where $1/\Delta$ is the "height" of both distributions, i.e., $\mathbb{P}(S_j^{\text{true}}=\mu_j^{\text{true}})=\mathbb{P}(S_j^{\text{false}}=\mu_j^{\text{false}})=1/\Delta$, v is the decay rate of tails. Let N_j^- (N_j^+) be the number of indices in $\mathcal{S}_j^{\text{false}}$ ($\mathcal{S}_j^{\text{true}}$). For ease of notations, we omit the subscript j in this proof since the detection is performed on each j individually.

Denote the order statistics of random variables in $\mathcal{S}^{\text{false}}$ by $S_{(1)}^{\text{false}},\cdots S_{(N^-)}^{\text{false}}$, where $S_{(1)}^{\text{false}}$ is the smallest order statistic and $S_{(N^-)}^{\text{false}}$ is the largest order statistic. The following lemma motivates the performance of the rank-based method.

Lemma 1. *The F_1-score of detecting corrupted labels in $\mathcal{N}_j$ by the rank-based method will be no less than $1-\alpha/N^-$ when the true probability $\mathbb{P}(Y=j|\widetilde{Y}=j)$ is known and $S_{(N^-)}^{false}<S_{(\alpha+1)}^{true}$.*

Lemma 1 connects the upper bound for the number of wrongly detected corrupted instances with order statistics. There are two cases that can cause detection errors:
Case-1:

$$0\leq\mu^{\text{true}}-S^{\text{true}}<\Delta \text{ and } 0\leq S^{\text{false}}-\mu^{\text{false}}<\Delta : \text{at most } \alpha \text{ errors when } S_{(N^-)}^{\text{false}}<S_{(\alpha+1)}^{\text{true}}.$$

and **Case-2**:

$$\mu^{\text{true}}-S^{\text{true}}\geq\Delta \text{ or } S^{\text{false}}-\mu^{\text{false}}\geq\Delta : \text{at most } \max(N_-,N_+) \text{ errors}$$

We analyze each case as follows.

Case-1: When Case-1 holds, we have

$$\mathbb{P}(\mu^{\text{true}} - S^{\text{true}} = x) \leq 1/\Delta, x \in [0, \Delta]$$

and

$$\mathbb{P}(S^{\text{false}} - \mu^{\text{true}} = x) \leq 1/\Delta, x \in [0, \Delta].$$

The above two inequalities show that the left tail of S^{true} and the right tail of S^{false} can be upper bounded by uniform distributions. Denote the corresponding uniform distribution by $U^{\text{true}} \sim \mathsf{Unif}(\mu^{\text{true}} - \Delta, \mu^{\text{true}})$ and $U^{\text{false}} \sim \mathsf{Unif}(\mu^{\text{false}}, \mu^{\text{false}} + \Delta)$.

With true $\mathbb{P}(Y = j|\widetilde{Y} = j)$, the detection errors only exist in the cases when the left tail of S^{true} and the right tail of S^{false} are overlapped. When the tails are upper bounded by uniform distributions, we have

$$\begin{aligned}
&\mathbb{P}(S^{\text{false}}_{(N^-)} < S^{\text{true}}_{(\alpha+1)}) \geq \mathbb{P}(U^{\text{false}}_{(N^-)} < U^{\text{true}}_{(\alpha+1)}) \\
=&\mathbb{P}\left(\left[U^{\text{false}} - \mu^{\text{false}}\right]_{(N^-)} + \mu^{\text{false}} < \left[U^{\text{true}} - (\mu^{\text{true}} - \Delta)\right]_{(\alpha+1)} + (\mu^{\text{true}} - \Delta)\right) \\
=&\mathbb{P}\left(\left[U^{\text{false}} - \mu^{\text{false}}\right]_{(N^-)} - \left[U^{\text{true}} - (\mu^{\text{true}} - \Delta)\right]_{(\alpha+1)} < \mu^{\text{true}} - \mu^{\text{false}} - \Delta\right).
\end{aligned}$$

Note

$$\left[U^{\text{false}} - \mu^{\text{false}}\right]_{(N^-)} \sim Beta(N^-, 1),$$

and

$$\left[U^{\text{true}} - (\mu^{\text{true}} - \Delta)\right]_{(\alpha+1)} \sim Beta(\alpha + 1, N^+ - \alpha),$$

where $Beta$ denotes the Beta distribution. Both variables are independent. Thus the PDF of the difference is

$$f(p) = \begin{cases} B(N^+ - \alpha, 1)p^{N^+ - \alpha}(1-p)^{\alpha+1}F(1, N^- + N^+, 1 - N^-; \alpha + 2; 1 - p, 1 - p^2)/A, & 0 < p \leq 1 \\ B(N^- -, N^+ - \alpha)(-p)^{N^+ - \alpha}(1+p)^{N^- + N^+ - \alpha - 1}F(N^+ - \alpha, -\alpha, N^- + N^+; N^- + N^+ - \alpha; 1 - p^2, 1 + p)/A, & -1 \leq p < 0 \\ B(N^- + \alpha, N^+ - \alpha)/A, & p = 0, \end{cases}$$

where $A = B(N^-, 1)B(\alpha + 1, N^+ - \alpha)$, $B(a, b) = \int_0^1 t^{a-1}(1-t)^{b-1}dt$

$$F(a, b_1, b_2; c; x, y) = \frac{\Gamma(c)}{\Gamma(a)\Gamma(c-a)} \int_0^1 t^{a-1}(1-t)^{c-a-1}(1-xt)^{-b_1}(1-yt)^{-b_2}\,\mathrm{d}t.$$

Therefore, we have

$$\mathbb{P}(S^{\text{false}}_{(N^-)} < S^{\text{true}}_{(\alpha+1)}) \geq \int_{-1}^{\mu^{\text{true}} - \mu^{\text{false}} - \Delta} f(p)dp.$$

Case-2 The other part, we have no more than $e^{-v} \cdot \max(N^-, N^+)$ corrupted instances that may have higher scores than one clean instance.

Wrap-up From the above analyses, we know, w.p. at least $\int_{-1}^{\mu^{\text{true}} - \mu^{\text{false}} - \Delta} f(p)dp$, there are at most $e^{-v}\max(N^-, N^+) + \alpha$ errors in detection corrupted instances. Note $\mathsf{Precision} = \mathsf{Recall}$ if we detect with the best threshold $N_j\mathbb{P}(Y = j|\widetilde{Y} = j)$. Therefore, the corresponding F_1-score would be at least $1 - \frac{e^{-v}\max(N^-, N^+) + \alpha}{N^-}$.

□