RELATIVE INSTANCE CREDIBILITY INFERENCE FOR LEARNING WITH NOISY LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

The existence of noisy labels usually leads to the degradation of generalization and robustness of neural networks in supervised learning. In this paper, we propose to use a simple theoretically guaranteed sample selection framework as a plug-in module to handle noisy labels.Specifically, we re-purpose a sparse linear model with incidental parameters as a unified Relative Instance Credibility Inference (RICI) framework, which will detect and remove outliers in the forward pass of each mini-batch and use the remaining instances to train the network. The credibility of instances is measured by the sparsity of incidental parameters, which can be ranked among other instances within each mini-batch to get a relatively consistent training mini-batch. The proposed RICI framework yields two variants that enjoy superior performance on the symmetric and asymmetric noise settings, respectively. We prove that our RICI can theoretically recover the clean data. Experimental results on several benchmark datasets and a real-world noisy dataset show the effectiveness of our framework.

1 INTRODUCTION

Deep learning has achieved remarkable success on many topics of supervised learning. The performance heavily relies on the quality of label annotation since deep models are susceptible to noisy labels and can easily memorize randomly labeled annotations (Zhang et al., 2017), leading to the degradation of generalization and robustness. In many real-world scenarios, it is expensive and difficult to obtain precise labels, exposing a realistic challenge for supervised deep models to learn with noisy data.

There is a large literature for this challenge from various perspectives, including modifying the network architectures (Xiao et al., 2015; Goldberger & Ben-Reuven, 2017; Chen & Gupta, 2015; Han et al., 2018a) or loss functions (Ghosh et al., 2017; Zhang & Sabuncu, 2018; Wang et al., 2019; Lyu & Tsang, 2020), or dynamically selecting clean data during training (Song et al., 2019; Lyu & Tsang, 2020; Han et al., 2018b; Jiang et al., 2018; Chen et al., 2019; Shen & Sanghavi, 2019; Yu et al., 2019; Nguyen et al., 2020). Particularly, the dynamic sample selection methods adopts the spirit of providing only clean data for the training. Such a spirit can form a 'virtuous' cycle between the noisy data elimination and network training: the elimination of noisy data can help the network training; and on the other hand, the improved network is empowered with better ability in picking up clean data. As this virtuous cycle evolves, the final performance can be significantly improved.

Many existing sample selection algorithms implicitly assume the samples with small loss (Han et al., 2018b) to be clean data. However, the small loss assumption relies on the inductive bias of the network that the majority pattern of each class is handled at the early training stage. This may fail when some wrong patterns were first memorized by the network, resulting in a small loss for such noisy data, especially when the pattern exists in a high portion of noisy data. Theoretically, there is no guarantee that these methods can consistently recover these clean data, thus leading to those failure cases.

To this end, from the statistical perspective we in this paper build up a simple sample selection framework, dubbed *Relative Instance Credibility Inference* (RICI), which has theoretical guarantees of consistently recovering clean data, and can be plugged into supervised methods with standard loss functions and network structures. Specifically, the RICI uses a sparse linear model with incidental parameters to detect and remove outlier samples in the forward pass of each mini-batch, and running the standard backward algorithm using the remaining clean data. Formally, we model the linear

relation between the feature-label pair (denoted as $(\boldsymbol{x}, \boldsymbol{y})$) received from the current network at each step:

$$\boldsymbol{y} = \boldsymbol{x}^\top \boldsymbol{\beta}^* + \boldsymbol{\gamma}^* + \boldsymbol{\varepsilon}, \tag{1}$$

where the incidental parameter $\boldsymbol{\gamma}^*$ models the non-random bias existed in the noisy label $\boldsymbol{y}$. Intuitively, the larger $\|\boldsymbol{\gamma}^*\|$ indicates more possibility for the $\boldsymbol{y}$ to be outlier/noisy data. After properly solving $\boldsymbol{\gamma}$ by optimizing the induced sparse linear regression problem, we only utilize the paired feature-label instance with $\boldsymbol{\gamma}(\boldsymbol{i}, \cdot) = 0$ to update the network.

Technically, we develop two variants of our RICI framework by using the labels either from original noisy training data (RICIN), or the noisy predictions of networks (RICIP) in Eq. (1). We analyze the statistical properties of these two variants, and make a theoretical understanding of our RICI framework. Furthermore, to further reveal the insights of our two variants, we introduce the settings of the symmetric and asymmetric noisy data. We find that both RICIN and RICIP can handle two types of noisy data. Interestingly RICIN can better handle the symmetric noise, whilst the RICIP can better deal with the asymmetric noisy data. Insightful, we give an explanation that RICIN can make noisy data elimination; and RICIP encourages the network learning by a curriculum learning strategy. We conduct extensive experiments to validate the effectiveness of our framework. The results show that our framework can better improve the performance of network learning than the competitors.

Contributions. We summarize our contributions as follows:

- We present a unified statistical approach, *i.e.*, RICI, to dynamically select the clean data under a general scenario.
- From the basic idea of RICI, we further propose two variants – RICIN and RICIP which can handle the symmetric and asymmetric noisy data.
- To the best of our knowledge, we make the first effort on theoretically guarantees of recovering the clean data from noisy dataset in the supervised manner.
- Our method can achieve the state-of-the-art results on a real-world noisy data challenge.

2 Related Work

2.1 Learning with noisy labels

We can roughly categorize LNL algorithms into two groups: architecture modification and sample selection. Architecture modification includes specific techniques for constructing robust network (Xiao et al., 2015; Goldberger & Ben-Reuven, 2017; Chen & Gupta, 2015; Han et al., 2018a), robust loss function (Ghosh et al., 2017; Zhang & Sabuncu, 2018; Wang et al., 2019; Lyu & Tsang, 2020), robust regularization (Tanno et al., 2019; Menon et al., 2020; Xia et al., 2021) against noisy labels. Sample selection aims to detect clean data and use the clean subset to train the neural network. We mainly review the sample selection algorithms.

Sample selection algorithms can be split into two parts: the selection algorithm for detecting clean data and the training algorithm for using the detected clean data. The selection criteria includes small loss (Shen & Sanghavi, 2019), gradient directions (Ren et al., 2018), disagreement within multiple networks (Yu et al., 2019), and some spatial properties in the training data (Wang et al., 2018; Lee et al., 2019; Wu et al., 2020). The motivation of using small loss criteria is shared by the curriculum learning (Zhou et al., 2021a), which aims to design a non-i.i.d. sequence of training instances to fit the network using easy data in the early stage and then gradually add the hard samples. Some algorithms (Veit et al., 2017; Ren et al., 2018) rely on the existence of an extra clean set to detect noisy data.

After detecting the clean data, the simplest strategy is to train the network using the clean data only or re-weight the data (Patrini et al., 2017) to eliminate the noise. Some algorithms (Li et al., 2020; Arazo et al., 2019) regard the detected noisy data as unlabeled data to fully exploit the distribution support of the training set in the semi-supervised learning manner. A commonly used strategy is using MixMatch (Berthelot et al., 2019) between the detected clean data and noisy data. There are also some studies of designing label-correction module (Xiao et al., 2015; Vahdat, 2017; Veit et al., 2017; Li et al., 2017; Tanaka et al., 2018; Yi & Wu, 2019) to further pseudo-labeling the noisy data to train the network. However, these approaches usually require a specific training pipeline with multiple networks or training rounds, resulting in extra memory and time consumption. On the

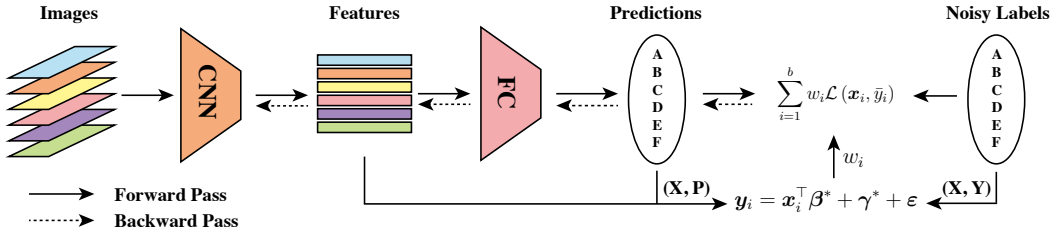


Figure 1: The illustration of our proposed framework. We aim to use a linear model to detect clean sample from noisy training set within each mini-batch, and generate a binary weight to train the network using only the clean subset indicated by RICI.

contrary, RICI is designed as a plug-in module to standard supervised training pipelines, leading to a simple but effective framework.

2.2 Incidental Parameters

The linear model with incidental parameters (Eq. (1)) is traditionally used by statisticians to learn a robust model against data-dependent noise (Neyman & Scott, 1948; Kiefer & Wolfowitz, 1956; Basu, 2011; Moreira, 2008; Fan et al., 2018). Fu et al. (2015) introduces the incidental parameter to solve the robust ranking problem. Recently Wang et al. (2020; 2021) utilize the linear model with incidental parameters as a self-taught learning algorithm combined with a linear classifier for few-shot learning and provide theoretical conditions for Eq. (1) to identify correctly pseudo-labeled instances. In this paper, we show that $\|\boldsymbol{\gamma}^*\|$ can be a general metric of credibility whose precise meaning is defined by the choice of $(\boldsymbol{x}, \boldsymbol{y})$. Further, we show that Eq. (1) can be designed as a plug-in module incorporated with a deep model to improve the performance with the existence of noisy labels. We also extend the formulation of Eq. (1) such that it will work in many situations when the standard algorithm fails.

3 Methodology

Problem Formulation. We are given a dataset of image-label pairs $\{(x_i, \bar{y}_i)\}_{i=1}^n$, where $x_i \in \mathcal{X} \subseteq \mathbb{R}^m, \bar{y}_i \in \mathcal{C} \subseteq \mathbb{R}, |\mathcal{C}| = c$. We assume that for each instance i, $\bar{y}_i$ is corrupted from the ground-truth category $y_i^\star$, where the corruption process is unknown. Our goal is predicting the ground-truth label $y^\star \in \mathcal{C}$ for any $x \in \mathcal{X}$. Denote $A(i,\cdot), A(\cdot,j), \|A\|_{\mathrm{F}} := \sqrt{\sum_{i,j} A^2(i,j)}$ as the i-th row, j-th column and the Frobenius norm of matrix A, respectively. Denote $\|a\|_1 := \sum_i |a_i|$ as the ℓ_1 norm of vector a.

Our framework, dubbed as *Relative Instance Credibility Inference* (RICI), is illustrated in Figure 1. For each mini-batch, the RICI selects the clean data for the network to train, by solving the incidental parameters γ in a sparse linear regression model. Specifically, as shown in Fig. 1, we use a sparse linear regression model to fit the feature-label pairs $\{x_i, y_i\}_i^b$ received from the current mini-batch, and solve the corresponding incidental parameters γ. We assume that the non-zero γ corresponds to the *inconsistent* data during fitting as in Fu et al. (2015). We thus take this sample as the outlier and set its weight $w = 0$ for this in the loss $\sum_{i=1}^b w_i \mathcal{L}(x_i, \bar{y}_i)$ that is then updated via backward propagation.

The rest of this section is organized as follows: we first introduce RICI, starting from the model assumption, *i.e.*, sparse linear model for the feature-label pair, and its induced loss and optimization in Sec. 3.1; in Sec. 3.2, we then discuss three variants of this framework under the symmetric and asymmetric noise settings and also the sparse regularization ℓ_q ($q < 1$) penalty in $\mathcal{L}$ to enforce the linear relationship between the feature and the label; finally, we provide the theoretical guarantees in recovering the clean dataset in Sec. 3.3.

3.1 Relative Instance Credibility Inference

For each mini-batch $\{(\boldsymbol{x}_i, \bar{y}_i)\}_{i=1}^b$, the key step of RICI is selecting clean subset from noisy data in each update. Under the assumption that the clean data takes a majority among all data (otherwise it is impossible to identify the clean data), we assume the following linear regression with incidental

parameters (Fan et al., 2018) for the feature-label pair in each forward pass:

$$\boldsymbol{Y} = \boldsymbol{X}^\top \boldsymbol{\beta}^* + \boldsymbol{\gamma}^* + \boldsymbol{\varepsilon}, \tag{2}$$

where ε is Gaussian noise; $\boldsymbol{Y} \in \mathbb{R}^{b\times c}$ and $\boldsymbol{X} \in \mathbb{R}^{b\times p}$ respectively denotes the one-hot encoded label matrix and the feature representation obtained as the output of the second last layer. The $\gamma^* \in \mathbb{R}^{b\times c}$ generates the noisy data, with larger $\|\gamma(i,\cdot)\|$ means more corruption the instance i is suffered. We denote $O := \{i : \|\boldsymbol{\gamma}^*(i,)\| \neq 0\}$ as the outlier sample set.

Remark. Note that Wang et al. (2020; 2021) adopts Eq. (2) in few-shot learning to gradually augment the training set, in which γ^* measures the credibility for the sample to be augmented.

To estimate O, we propose to solve the following sparse linear regression:

$$\underset{\boldsymbol{\beta},\boldsymbol{\gamma}}{\operatorname{argmin}} \frac{1}{2} \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{\gamma}\|_{\mathrm{F}}^2 + \lambda \sum_{i=1}^{b} \|\boldsymbol{\gamma}_i\|_1 , \tag{3}$$

To simplify the optimization, we substitute the closed-form solution for $\boldsymbol{\beta}$ (*i.e.*, $\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^\dagger \boldsymbol{X}^\top (\boldsymbol{Y} - \boldsymbol{\gamma})$ with γ fixed) into Eq. (3). To ensure that $\hat{\boldsymbol{\beta}}$ is identifiable, we apply PCA on $\boldsymbol{X}$ to make $p \ll b$ so that the $\boldsymbol{X}$ has full-column rank. Denote $\tilde{\boldsymbol{X}} = \boldsymbol{I} - \boldsymbol{X}\left(\boldsymbol{X}^\top \boldsymbol{X}\right)^\dagger \boldsymbol{X}^\top, \tilde{\boldsymbol{Y}} = \tilde{\boldsymbol{X}}\boldsymbol{Y}$, the Eq. (3) is transformed into

$$\underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \frac{1}{2} \left\|\tilde{\boldsymbol{Y}} - \tilde{\boldsymbol{X}}\boldsymbol{\gamma}\right\|_{\mathrm{F}}^2 + \lambda \sum_{i=1}^{b} \|\boldsymbol{\gamma}_i\|_1 , \tag{4}$$

which is a standard sparse linear regression for $\boldsymbol{\gamma}$. We use Glmnet (Simon et al., 2013) to generate the solution path of $\boldsymbol{\gamma}$ with respect to the λ. Since earlier selected instance is more possible to be noisy, we rank all samples as the descendent order of their selecting time defined as:

$$C_i = \sup\{\lambda : \boldsymbol{\gamma}_i(\lambda) \neq 0\}. \tag{5}$$

A large C_i means that the $\boldsymbol{\gamma}_i$ is earlier selected if we run from $\lambda = \infty$ to 0. We then select the samples that are less than $\alpha\%$ quantile of $\{C_i\}$, as *relatively* clean data compared with other data in current mini-batch.

3.2 Learning with RICI

In this section, we discuss two variants of the RICI framework in Eq. (3) under the symmetric and asymmetric noise setting, in which the noisy data is respectively randomly distributed and only restricted to some particular classes, respectively. Consider the digital recognition example, the digit "7" can be randomly mislabeled as other digits from "0" to "9" in the symmetric noise setting; or mislabeled as "1" that can expose similar patterns in the asymmetric noise setting. Therefore, the noisy data in the asymmetric setting has a large overlapping with hard samples, which hence motivated a stream of curriculum learning methods such as Zhou et al. (2021b) in this setting. In the following, we consider the label $\boldsymbol{Y}$ in Eq. (3) as noisy label $\bar{\boldsymbol{Y}}$ and prediction one-hot encoded label $\boldsymbol{P} \in \mathbb{R}^{b\times c}$, as two variants respectively corresponding to symmetric and asymmetric setting.

- **RICIN for Symmetric Noise.** We take the noisy label $\bar{\boldsymbol{Y}}$ as $\boldsymbol{Y}$ in Eq. (3) and denote it as the *RICI Noise* (RICIN). In this regard, the clean part of the label (*i.e.*, $\boldsymbol{x}^\top \boldsymbol{\beta}^*$) corrupted by the non-zero $\boldsymbol{\gamma}$. Hence, the non-zero $\boldsymbol{\gamma}_i$ implies the existence of non-random noise in the label $\bar{\boldsymbol{Y}}_i$, making the instance i as a candidate for to be the noisy data.
- **RICIP for Asymmetric Noise.** Under this setting, it is reasonable to take the noisy data as the hard cases. Therefore, we can adopt the curriculum learning strategy, in which we gradually pick up from easy to hard samples. To achieve this goal, we take the $\boldsymbol{Y}$ as $\boldsymbol{P}$, *i.e.*, one-hot encoded vector of obtained from $\operatorname{softmax}(\boldsymbol{X}\boldsymbol{W}_{\mathrm{fc}})$, where $\boldsymbol{W}_{\mathrm{fc}}$ denotes the weight in the fully-connected layers (for simplicity we ignore the bias term in the formulation) with the element on $\operatorname{argmax}_c \operatorname{softmax}(\boldsymbol{X}\boldsymbol{W}_{\mathrm{fc}})$ being 1 and others being 0. In this regard, the non-zero $\boldsymbol{\gamma}$ correspond to the samples that are hard to fit, *i.e.*, hard samples. Thus we exclude them to use easy sample to train the network. We denote the RICI with $\boldsymbol{Y} = \boldsymbol{P}$ in Eq. (3) as *RICI Predict* (RICIP).

- **RICIC: Combination of RICIN and RICIP.** To utilize both the learning efficiency from RICIP and the ability of selecting noisy data from RICIN, we propose the *RICI Concatenation* (RICIC), which take the $\boldsymbol{Y}$ as the concatenation of $\bar{\boldsymbol{Y}}$ and $\boldsymbol{P}$:

$$\underset{\boldsymbol{\gamma}}{\operatorname{argmin}} \frac{1}{2} \left\| \begin{pmatrix} \boldsymbol{Y} \\ \boldsymbol{P} \end{pmatrix} - \begin{pmatrix} \tilde{\boldsymbol{X}} \\ \tilde{\boldsymbol{X}} \end{pmatrix} \boldsymbol{\gamma} \right\|_{\mathrm{F}}^2 + \lambda \sum_{i=1}^{b} \|\boldsymbol{\gamma}_i\|_1 , \tag{6}$$

After estimating the clean set O, we set $w_i = 1$ for $i \notin O$ and $= 0$ otherwise. Then we update the network parameter with loss $\sum_i w_i \mathcal{L}(x_i, \bar{y}_i)$, with $\mathcal{L}(\cdot)$ usually adopted as cross-entropy loss. Note that we assume in Eq. (2) that the one-hot encoded label is linearly related to the feature $\boldsymbol{X}$; however, in practice the network prediction is obtained via the soft-max function on the $\boldsymbol{X}\boldsymbol{W}_{\mathrm{fc}}$. To reduce this gap, we append a ℓ_q ($q < 1$) penalty on the cross entropy loss, which encourages the linear relationship between $\boldsymbol{X}$ and one-hot encoded vector $\boldsymbol{Y}$:

$$\mathcal{L}(\boldsymbol{x}, \bar{y}) = \mathcal{L}_{\mathrm{CL}}(\boldsymbol{x}, \bar{y}) + \lambda \|\boldsymbol{x}^\top W_{\mathrm{fc}}\|_q, \tag{7}$$

where $q < 1$ and $\mathcal{L}_{\mathrm{CL}}$ denotes the cross-entropy loss. Note that the $\|\boldsymbol{x}^\top W_{\mathrm{fc}}\|_q$ enforces the $\boldsymbol{x}^\top W_{\mathrm{fc}}$ to approximately be one-hot encoded vector as long as q is small enough. We use $q = 0.2$ here.

Remark. Note that our feature selection module, *i.e.*, Eq. (3) is orthogonal to any choice of the loss $\mathcal{L}$. We will show in the experimental part that our RICI can improve over other choices of $\mathcal{L}$. Furthermore, RICI can also be regarded as a loss adjustment algorithm since we do not require any other modifications to the network structure or training process, except for the 0-1 re-weight and ℓ_q penalty.

3.3 Identifiability of RICI

In this section, we provide the identifiability result that the Eq. (4) can recover the oracle support set O. Our analysis is built upon the model selection consistency result of LASSO (Zhao & Yu, 2006; Wainwright, 2009). Specifically, we first vectorize $\boldsymbol{Y}, \boldsymbol{\gamma}$ in (4) as $\vec{\boldsymbol{y}}, \vec{\boldsymbol{\gamma}}$ and the Eq. (4) turns to

$$\underset{\vec{\boldsymbol{\gamma}}}{\operatorname{argmin}} \frac{1}{2} \left\| \vec{\boldsymbol{y}} - \mathring{\boldsymbol{X}} \vec{\boldsymbol{\gamma}} \right\|_2^2 + \lambda \left\| \vec{\boldsymbol{\gamma}} \right\|_1 , \tag{8}$$

where $\mathring{\boldsymbol{X}} = I_c \otimes \tilde{\boldsymbol{X}}$ with $\otimes$ denoting the Kronecker product operator. Denote $S := \operatorname{supp}(\vec{\boldsymbol{\gamma}}^*)$, then it sufficient for the recovery of noisy set O to recover S. We further denote $\mathring{\boldsymbol{X}}_S(\mathring{\boldsymbol{X}}_{S^c})$ as the column vectors of $\mathring{\boldsymbol{X}}$ whose indexes are in $S(S^c)$ and $\mu_{\mathring{\boldsymbol{X}}} = \max_{i \in S^c} \|\mathring{\boldsymbol{X}}\|_2^2$. Then we have

Theorem 1 (Idenifiability (Wang et al., 2021))**.** *Assume that:*
C1, Restricted eigenvalue: $\lambda_{\min}(\mathring{\boldsymbol{X}}_S^\top \mathring{\boldsymbol{X}}_S) = C_{\min} > 0$*;*
C2, Irrepresentability: $\exists \eta \in (0, 1], \|\mathring{\boldsymbol{X}}_{S^c}^\top \mathring{\boldsymbol{X}}_S (\mathring{\boldsymbol{X}}_S^\top \mathring{\boldsymbol{X}}_S)^{-1}\|_\infty \leq 1 - \eta$*;*
C3, Large Signal-to-Noise Ratio (SNR): $\vec{\boldsymbol{\gamma}}^*_{\min} := \min_{i \in S} |\vec{\boldsymbol{\gamma}}_i^*| > h(\lambda, \eta, \mathring{\boldsymbol{X}}, \vec{\boldsymbol{\gamma}}^*)$*;*
where $h(\lambda, \eta, \mathring{\boldsymbol{X}}, \vec{\boldsymbol{\gamma}}^*) = \frac{\lambda \eta}{\sqrt{C_{\min} \mu_{\mathring{\boldsymbol{X}}}}} + \lambda \|(\mathring{\boldsymbol{X}}_S^\top \mathring{\boldsymbol{X}}_S)^{-1} \operatorname{sign}(\vec{\boldsymbol{\gamma}}_S^*)\|_\infty$ *and* $\|\boldsymbol{A}\|_\infty := \max_i \sum_j |A_{i,j}|$.
Let $\lambda \geq \frac{2\sigma\sqrt{\mu_{\mathring{\boldsymbol{X}}}}}{\eta} \sqrt{\log cn}$*. Then with probability greater than* $1 - 2(cn)^{-1}$*, model (8) has a unique solution* $\hat{\vec{\boldsymbol{\gamma}}}$ *such that: 1) If C1 and C2 hold,* $\hat{O} \subseteq O$*;2) If C1, C2 and C3 hold,* $\hat{O} = O$*.*

In our scenario, the C1 can be satisfied since the S is much smaller than S^c (we assume that the clean data is the majority). The C3 implies that we can only select the signal once the SNR is large enough. As an almost necessary condition, the C2 is the key assumption for the support set S to be identified. However, this assumption may not be easy to satisfy. To amend this problem, we propose to precondition the matrix pairs $(\mathring{\boldsymbol{X}}, \vec{\boldsymbol{y}})$ by left multiplying a matrix $\boldsymbol{F}$ such that $\boldsymbol{F}\mathring{\boldsymbol{X}}$ can automatically satisfies the irrepresentability condition. One possible property for satisfying C2 is to orthogonalize $\mathring{\boldsymbol{X}}$ such that for each column pair $(i, j), i \neq j$, we have $\mathring{\boldsymbol{X}}_i^\top \mathring{\boldsymbol{X}}_j = 0$. Then we have $\mathring{\boldsymbol{X}}_{S^c}^\top \mathring{\boldsymbol{X}}_S (\mathring{\boldsymbol{X}}_S^\top \mathring{\boldsymbol{X}}_S)^{-1} = \boldsymbol{O}$ and the irrepresentability condition is satisfied. To achieve the orthogonal property, we use the Puffer transformation (Jia & Rohe, 2015), where we first calculate the SVD decomposition for $\mathring{\boldsymbol{X}}$ such that $\mathring{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}$, and we define $\boldsymbol{F} = \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^\top$ such that

$$\boldsymbol{F}\mathring{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{D}^{-1}\boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{D}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{V}, \tag{9}$$

where $(\boldsymbol{F}\mathring{\boldsymbol{X}})^\top \boldsymbol{F}\mathring{\boldsymbol{X}} = \boldsymbol{V}^\top \boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{V} = \boldsymbol{I}$, leading to the orthogonal property.

Theorem 2 (Preconditioned Identifiabilty)**.** *Define*

$$\tilde{\gamma} = \underset{\vec{\gamma}}{\operatorname{argmin}} \frac{1}{2} \left\| \boldsymbol{F}\vec{\boldsymbol{y}} - \boldsymbol{F}\mathring{\boldsymbol{X}}\vec{\gamma} \right\|_2^2 + \lambda \left\| \vec{\gamma} \right\|_1, \tag{10}$$

Let $\lambda \geq \frac{2\sigma\sqrt{\mu_{\mathring{\boldsymbol{X}}}}}{\eta}\sqrt{\log cn}$*. If C1 and C3 hold, then* $\hat{O} = O$ *with probability greater than* $1 - 2(cn)^{-1}$.

The proof is given in the Appendix A. Note that the Puffer transformation will transfer the standard linear regression model (8) into a Pre-conditioned lasso model, where the random noise is dependent on each other and often get a higher variance. Hence in practice we should balance the benefit of automatically satisfying the irrepresentability condition and costs of introducing dependent noise of higher variance.

Table 1: Test accuracies on several benchmark datasets with different settings. Every result is averaged over 3 different random runs. The best result is **boldfaced**. Results of competitors on MNIST and CIFAR10 are reported in (Zhou et al., 2021c). Results on SVHN are reproduced by ourselves using the provided code. CnFm implies the network includes n convolutional layers followed by m fully-connected layers.

Dataset	Method	Clean	Symmetric Noise Rate				Asymmetric Noise Rate		
			0.2	0.4	0.6	0.8	0.2	0.3	0.4
MNIST (C2F2)	CE	99.15	91.62	73.98	49.36	22.66	94.56	88.81	82.27
	FL	99.13	91.68	74.54	50.39	22.65	94.25	89.09	82.13
	GCE	99.27	98.86	97.16	81.53	33.95	96.69	89.12	81.51
	SCE	99.23	98.92	97.38	88.83	48.75	98.03	93.68	85.36
	NLNL	98.85	98.33	97.80	96.18	86.34	98.35	97.51	95.84
	APL	99.34	99.14	98.42	95.65	72.97	98.89	96.93	91.45
	SR	99.33	**99.22**	99.16	98.85	98.06	**99.27**	99.24	**99.23**
	RICIN	98.59	98.92	99.11	**99.09**	**98.64**	98.66	98.72	98.65
	RICIP	99.27	99.18	**99.17**	98.96	22.35	99.26	**99.25**	99.19
	RICIC	98.51	98.99	99.07	99.07	95.52	98.64	98.64	98.64
CIFAR10 (C6F2)	CE	90.48	74.68	58.26	38.70	19.55	83.32	79.32	74.67
	FL	89.82	73.72	57.90	38.86	19.13	83.37	79.33	74.28
	GCE	89.59	87.03	82.66	67.70	26.67	85.93	80.88	74.29
	SCE	91.61	87.10	79.67	61.35	28.66	86.20	81.38	75.16
	NLNL	90.73	73.70	63.90	50.68	29.53	84.74	81.26	76.97
	APL	89.17	86.98	83.74	76.02	46.69	86.50	83.34	77.14
	SR	90.06	87.93	84.86	78.18	51.13	87.70	85.63	79.29
	RICIN	84.65	86.85	**86.49**	81.86	**54.05**	85.23	84.22	81.71
	RICIP	91.05	**88.70**	86.04	79.39	37.80	**89.20**	**87.42**	**84.97**
	RICIC	85.49	87.35	86.33	**81.98**	53.44	85.43	84.17	80.48
SVHN (WRN16)	CE	96.80	90.67	82.60	67.78	68.04	91.58	87.03	81.83
	FL	96.77	89.83	81.90	68.00	67.87	94.15	92.76	86.94
	GCE	96.81	91.07	82.47	68.48	66.56	91.06	87.25	81.48
	SCE	96.97	**96.54**	**95.71**	93.55	81.78	96.51	95.35	90.70
	SR	96.81	96.37	95.61	93.05	19.59	96.37	95.76	95.19
	RICIN	84.71	87.27	91.77	93.10	81.67	82.59	83.18	82.13
	RICIP	96.87	96.15	95.61	93.85	**83.77**	**96.73**	**96.35**	**96.46**
	RICIC	93.78	94.08	94.50	**94.10**	82.53	92.62	92.94	92.17

4 EXPERIMENTS

Setup. We investigate the effectiveness of RICI on several benchmark datasets including MNIST (LeCun et al., 1998), CIFAR10 (Krizhevsky et al., 2009), SVHN (Netzer et al., 2011) and a real-world dataset ANIMAL10 (Song et al., 2019). We consider two types of noisy labels for MNIST, CIFAR10, and SVHN: (i) Symmetric noise: Every class is corrupted uniformly with all other labels; (ii) Asymmetric noise: Labels are corrupted by similar (in pattern) classes. The

ANIMAL10 is published with mislabeling (the ratio is 8%) and the corruption process and noise type in ANIMAL10 are unknown. Thus, the ANIMAL10 can be regarded as a real-world challenge.

Backbones. To make a fair comparison with other algorithms, we use different backbones for different datasets. For MNIST, we use two convolutional layers followed by two fully-connected layers, denoted as *C2F2*. For CIFAR10, a 6-layer CNN followed by two fully-connected layers is utilized, denoted as *C6F2*. WideResNet-16-8 (Zagoruyko & Komodakis, 2016) is used for SVHN, and for ANIMAL10 we use VGG19-BN (Simonyan & Zisserman, 2015) as our backbone.

Hyperparameter setting. We use SGD to train all the networks with momentum 0.9, except for ANIMAL10 we do not use momentum. For SVHN and ANIMAL10 the learning rate is decayed after certain epochs, while for others we use the cosine learning rate decay algorithm. The initial learning rate is set as 0.1 for ANIMAL10 and 0.01 for others. The weight decay is set as 1e-3,1e-4,5e-4,1e-3 for MNIST, CIFAR10, SVHN, and ANIMAL10, respectively. We use a batch size of 128. We use random crop and random horizontal flip as augmentation strategies for CIFAR10, SVHN, and ANIMAL10. The network is trained for 50 epochs for MNIST, 120 epochs for CIFAR10, and 160 epochs for SVHN and ANIMAL10. The coefficient λ of the sparse penalty is initialized as 4 for MNIST and 1.2 for others, and is increased by multiplying 2 for MNIST, 1.03 for CIFAR10 and ANIMAL10, 1.02 for SVHN, and 1.014 for the experiments of symmetric noise rate 0.8 on CIFAR10. We simply select half of the training data in all of our experiments.

4.1 Evaluation on Synthetic Label Noise

Competitors. We first utilize the cross-entropy loss (CE) as a baseline algorithm. Another effective loss function Focal Loss (FL) (Lin et al., 2017) is also compared. Some refined algorithms for CE loss, GCE (Zhang & Sabuncu, 2018) and SCE (Wang et al., 2019), are also compared. NLNL (Kim et al., 2019) utilizes complementary labels to against the noise. APL (Ma et al., 2020) combines robust active and passive loss to train the network. SR (Zhou et al., 2021c) utilizes the sparse regularization combined with the feature normalization and temperature scaling method to train the network. To make a fair comparison, we use the same backbone and hyper-parameters for all methods.

The results are shown in Table 1. Our algorithm enjoys comparable or better performance in all settings. Specifically, when the noisy rate is not large (*i.e.*, asymmetric noise where the noise only occurs in a subset of classes and low noisy rate in symmetric noise), the noisy data can have a minor influence on the training, leading to the superior performance of RICIP which select easy samples to train. With symmetric noise of high noise rate, the RICIN performs more stable than RICIP, since the selected easy samples of RICIP no longer correspond to clean data. As a combined version, the RICIC can bring additional benefits in some scenarios. In practice, the RICIC is suggested as a starting baseline for the specific noisy label problem when the noise type is unknown.

4.2 Evaluation on Real-World Noisy Dataset

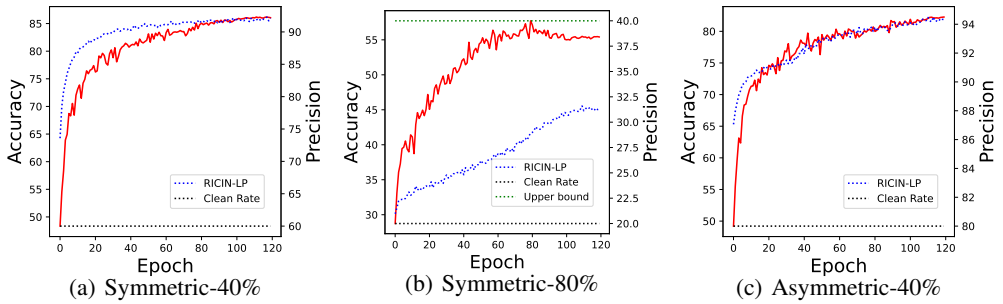
Table 2: Test accuracies on ANIMAL10. Every result is averaged over 3 different random runs. The best result are **boldfaced**.

CE	Nested	CED	SELFIE	PLC	NCT	RICIN	RICIP	RICIC
79.4	81.3	81.3	81.8	83.4	84.1	77.14	**84.74**	76.30

In this section, we compare RICI with other methods in a real-world noisy dataset, ANIMAL10.

Competitors. We compare with the baseline of directly training with cross-entropy loss (CE), as well as previous works including Nested(ND), CE + Dropout (CED), SELFIE (Song et al., 2019), PLC (Zhang et al., 2021), and NestedCoTeaching (NCT) (Chen et al., 2021).

Results are shown in Table 2, where the results of CE and SELFIE is reported in (Song et al., 2019), the results of ND, CED, and NCT is reported in (Chen et al., 2021), while the result of PLC is reported in their paper. Our algorithm RICIP enjoys superior performance to all the competitors, showing the ability of handling real-world challenge. Since the noise rate is low (8%), the other two variants eliminate many clean data due to the strategy of selecting only half of the training data, resulting in the inferior performance of RICIN and RICIC.



(a) Symmetric-40% (b) Symmetric-80% (c) Asymmetric-40%

Figure 2: Accuracy and Label precision of RICIN under different noise scenarios. The red line is the accuracy of RICIN, while the dotted line is the label precision.

4.3 Empirical Analysis of RICI

Precision of sample selection. Besides accuracy, another metric to test the capacity of a sample selection algorithm is *precision*: the ratio of true clean labels in the selected instances. In this part, we check the precision of RICIN to show the sample selection effectiveness. We conduct our experiments on the symmetric noise rate of 0.4 and 0.8, as well as asymmetric noise rate of 0.4. Results are shown in Figure 2. RICIN enjoys a monotonically increasing label precision, leading to a better training environment than the standard noisy dataset. When the training process ends, almost all the selected training data is guaranteed to be clean data (92.16% in the symmetric-40% setting and 94.33% in the asymmetric-40% setting). Note that in the symmetric-80% setting, due to the strategy of selecting half of the training data, the upper bound of the precision is 40%, as illustrated. In this high noise rate scenario, RICIN can still achieve the precision of 31.30%, which means that 73.27% of the clean training instances are detected by our algorithm. The above results consistently show the effectiveness of RICIN.

Ablation study of modules in RICI. To verify the effectiveness of each module in our framework, we conduct an ablation study on CIFAR10 with 40% symmetric noise rate. Specifically, the "CE" denotes vanilla cross entropy method; the "CE + RICIN" means the cross-entropy loss for $\mathcal{L}$ with Eq. equation 4 to identify $\{w_i\}$ in each forward pass; the "CE + ℓ_q" means the Eq. (7) for $\mathcal{L}$; and the "Full" denotes our RICI method with all components. As shown in Table 3, simply using our framework will lead to better performance compared with the standard CE loss. With the additional ℓ_q norm appended on the loss, the linear relationship between the feature and the label is enforced so that the RICI can perform better in selecting clean data.

Table 3: Accuracy of using different modules in RICIN.

Model	Accuracy
CE	58.36
CE + RICIN	67.22
CE + ℓ_q	79.71
Full	86.49

Plug-in property of RICI. To show the effectiveness of our sample selection mechanism in Eq. (4), we also conduct RICI with other choices of the loss function. Here we consider FL and GCE. As shown in Table 4, our sample selection mechanism can achieve a large improvement. Therefore, we believe that our RICI framework is a plug-in module and can be applied to different loss functions. Besides, our experiments in Table 1 and Table 2 are conducted on different backbones, which shows that our RICI framework can be incorporated with different network architectures. Thus, our RICI framework can be used as a plug-in module that does not require specific loss function or network architecture.

Table 4: Accuracy of RICIN with different loss functions.

Model	Accuracy
FL	57.90
FL + RICIN	86.53
GCE	82.66
GCE + RICIN	86.29

Influence of the Puffer transformation. As shown in theorems 1,2, we propose to use the Puffer transformation to ensure the identifiability of RICI in the scenario where the irrepresentability condition is not satisfied by the standard formulation at a cost of making noise dependent and having larger variance. Specifically, we conduct experiments on CIFAR10 with symmetric noise of 40% and test the performance of RICIN and RICIP with or without using the Puffer transformation.

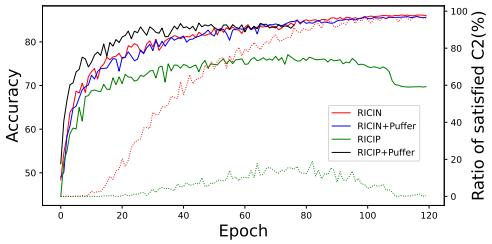


Figure 3: Accuracies (solid line) and ratios of mini-batches satisfying irrepresentability condition (dotted line) for each epoch.

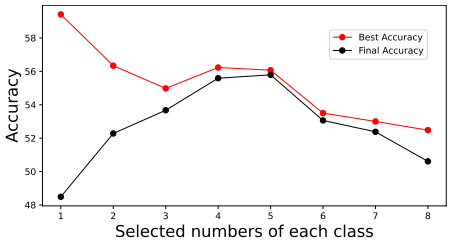


Figure 4: Best and final accuracies of RICIN running with selecting different number of training data.

Results are shown in Figure 3. Note that for variants using the Puffer transformation, the ratio is 100% and not visualized in the figure. The original formulation of RICIP cannot satisfy the irrepresentability condition and hence cannot recover the inconsistent data, resulting in an inferior performance. The preconditioned variant, RICIP+Puffer, enjoys a significant improvement thanks to the training environment for satisfying the irrepresentability condition. For RICIN, as the training process goes, the ratio of the satisfied mini-batches is gradually increased and converged to almost 100%, hence a preconditioning environment is not required since it will introduce extra dependent noise. Thus in our experiments, we use the Puffer transformation for RICIP, while do not use it for RICIN.

Influence of select ratio. In our experiments, we simply select half of the training data to train the network. It is desirable to investigate how does the number of selected instances influences the training process. We conduct experiments of RICIN on CIFAR10 with symmetric noise rate of 0.8. We use a batch size of 256, where the expectation of clean data in each mini-batch is $256 * 0.2/10 \approx 5$. It can be found that the best selection ratio is near the clean ratio in the training set. Hence a better selection strategy may be designed based on the estimation of the noise ratio in the training set. We leave it as a future work since in this paper we mainly propose the plug-in sample selection framework.

Influence of ℓ_q. In this part, we investigate the inlfuence of ℓ_q norm in our framework. We run with a sequence of q from 0.05 to 1, as illustrated in Figure 5. In general, a smaller q encourages the linear relation as expected by our framework, while too small q will damage the representation capacity of the network. Thus, a convex accuracy curve exists when we test with different ℓ_q, suggesting a choice of $q = 0.2$ to be the best to balance the linear relation requirement and the representation capacity. Hence in our experiments we use $q = 0.2$.

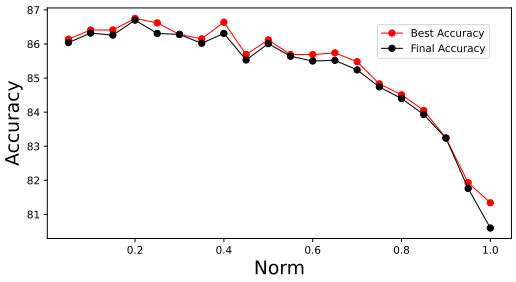


Figure 5: Accuracies of RICIN with different ℓ_q.

5 Conclusion and Future Work

In this paper, we propose a statistical plug-in sample selection framework, dubbed as Relative Instance Credibility Inference (RICI) so select clean data with theoretical guarantees. Specifically, we re-purpose a sparse linear model with incidental parameters, whose sparsity can be induced as a general metric for the relative credibility of instances within a mini-batch. Then one can rank and select the most consistent training data to train the network. We provide theoretical conditions to guarantee the identifiability of RICI to recover the oracle inconsistent set. Experiments on several synthetic benchmark datasets and a real-world dataset show the effectiveness of our framework. Since we organize our framework as a plug-in module for a standard supervised training pipeline, some modules are not specifically designed and maybe the future work of our framework, including a noise rate estimation algorithm to guide the number of selected instances and a combination with semi-supervised algorithms to further exploit the support of detected noisy data.

REFERENCES

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning*, pp. 312–321. PMLR, 2019. 2.1

Debabrata Basu. On the elimination of nuisance parameters. In *Selected Works of Debabrata Basu*. 2011. 2.2

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2.1

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pp. 1062–1070. PMLR, 2019. 1

Xinlei Chen and Abhinav Gupta. Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1431–1439, 2015. 1, 2.1

Yingyi Chen, Xi Shen, Shell Xu Hu, and Johan AK Suykens. Boosting co-teaching with compression regularization for label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2688–2692, 2021. 4.2

Jianqing Fan, Runlong Tang, and Xiaofeng Shi. Partial consistency with sparse incidental parameters. *Statistica Sinica*, 28:2633, 2018. 2.2, 3.1

Yanwei Fu, Timothy M Hospedales, Tao Xiang, Jiechao Xiong, Shaogang Gong, Yizhou Wang, and Yuan Yao. Robust subjective visual property prediction from crowdsourced pairwise labels. 2015. 2.2, 3

Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017. 1, 2.1

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2017. 1, 2.1

Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor W Tsang, Ya Zhang, and Masashi Sugiyama. Masking: a new perspective of noisy supervision. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 5841–5851, 2018a. 1, 2.1

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, 2018b. 1

Jinzhu Jia and Karl Rohe. Preconditioning the lasso for sign consistency. *Electronic Journal of Statistics*, 9(1):1150–1172, 2015. 3.3

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. 1

Jack Kiefer and Jacob Wolfowitz. Consistency of the maximum likelihood estimator in the presence of infinitely many incidental parameters. *The Annals of Mathematical Statistics*, 1956. 2.2

Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. Nlnl: Negative learning for noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 101–110, 2019. 4.1

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4

Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. Robust inference via generative classifiers for handling noisy labels. In *International Conference on Machine Learning*, pp. 3763–3772. PMLR, 2019. 2.1

Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020. 2.1

Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1910–1918, 2017. 2.1

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017. 4.1

Yueming Lyu and Ivor W Tsang. Curriculum loss: Robust learning and generalization against label corruption. 2020. 1, 2.1

Xingjun Ma, Hanxun Huang, Yisen Wang, Simone Romano, Sarah Erfani, and James Bailey. Normalized loss functions for deep learning with noisy labels. In *International Conference on Machine Learning*, pp. 6543–6553. PMLR, 2020. 4.1

Aditya Krishna Menon, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations*, 2020. 2.1

Marcelo Moreira. A maximum likelihood method for the incidental parameter problem. Technical report, 2008. 2.2

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 4

Jerzy Neyman and Elizabeth L Scott. Consistent estimates based on partially consistent observations. *Econometrica: Journal of the Econometric Society*, 1948. 2.2

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. 2020. 1

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. 2.1

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343. PMLR, 2018. 2.1

Yanyao Shen and Sujay Sanghavi. Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning*, pp. 5739–5748. PMLR, 2019. 1, 2.1

Noah Simon, Jerome Friedman, and Trevor Hastie. A blockwise descent algorithm for group-penalized multiresponse and multinomial regression. *arXiv preprint arXiv:1311.6529*, 2013. 3.1

K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. 2015. 4

Hwanjun Song, Minseok Kim, and Jae-Gil Lee. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*, pp. 5907–5915. PMLR, 2019. 1, 4, 4.2

Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5552–5560, 2018. 2.1

Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11244–11253, 2019. 2.1

Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. *Advances in Neural Information Processing Systems*, 30:5596–5605, 2017. 2.1

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 839–847, 2017. 2.1

Martin J Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1 -constrained quadratic programming (lasso). *IEEE transactions on information theory*, 2009. 3.3

Yikai Wang, Chengming Xu, Chen Liu, Li Zhang, and Yanwei Fu. Instance credibility inference for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12836–12845, 2020. 2.2, 3.1

Yikai Wang, Li Zhang, Yuan Yao, and Yanwei Fu. How to trust unlabeled data instance credibility inference for few-shot learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 2.2, 3.1, 1, A, 3, 4

Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. Iterative learning with open-set noisy labels. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8688–8696, 2018. 2.1

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/CVF International Conference on Computer Vision*, pp. 322–330, 2019. 1, 2.1, 4.1

Pengxiang Wu, Songzhu Zheng, Mayank Goswami, Dimitris N Metaxas, and Chao Chen. A topological filter for learning with label noise. *Advances in neural information processing systems*, 33, 2020. 2.1

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. 2.1

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. 1, 2.1

Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7017–7025, 2019. 2.1

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pp. 7164–7173. PMLR, 2019. 1, 2.1

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*. British Machine Vision Association, 2016. 4

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. 2017. 1

Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. Learning with feature-dependent label noise: A progressive approach. In *International Conference on Learning Representations*, 2021. 4.2

Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*, 2018. 1, 2.1, 4.1

Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine learning research*, 7(Nov):2541–2563, 2006. 3.3

Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Curriculum learning by optimizing learning dynamics. In *International Conference on Artificial Intelligence and Statistics*, pp. 433–441. PMLR, 2021a. 2.1

Tianyi Zhou, Shengjie Wang, and Jeff Bilmes. Robust curriculum learning: From clean label detection to noisy label self-correction. In *International Conference on Learning Representations*, 2021b. 3.2

Xiong Zhou, Xianming Liu, Chenyang Wang, Deming Zhai, Junjun Jiang, and Xiangyang Ji. Learning with noisy labels via sparse regularization. In *ICCV*, 2021c. 1, 4.1

A Appendix

Wang et al. (2021) proved Theorem 1 by a combination of a proposition with a lemma. Here we first introduce the proposition and the lemma, and then extend them in the dependent noise setting.

Proposition 3 (Wang et al. (2021))**.** *Assume that* $\mathring{\boldsymbol{X}}^\top \mathring{\boldsymbol{X}}$ *is invertible. If*

$$\left\| \lambda \mathring{\boldsymbol{X}}_{S^c}^\top \mathring{\boldsymbol{X}}_S \left(\mathring{\boldsymbol{X}}_S^\top \mathring{\boldsymbol{X}}_S \right)^{-1} \hat{\boldsymbol{v}}_S + \mathring{\boldsymbol{X}}_{S^c}^\top \left(\boldsymbol{I} - \boldsymbol{I}_S \right) \mathring{\boldsymbol{X}} \boldsymbol{\varepsilon} \right\|_\infty < \lambda \tag{11}$$

holds for all $\hat{\boldsymbol{v}}_S \in [-1,1]^S$*, where* $\boldsymbol{I}_S = \mathring{\boldsymbol{X}}_S \left(\mathring{\boldsymbol{X}}_S^\top \mathring{\boldsymbol{X}}_S \right)^{-1} \mathring{\boldsymbol{X}}_S^\top$*, then* $\hat{S} = \operatorname{supp}\left(\hat{\vec{\gamma}}\right) \subseteq \operatorname{supp}\left(\vec{\gamma}^*\right) = S$*. Moreover, if* $\operatorname{sign}\left(\hat{\vec{\gamma}}_S\right) = \operatorname{sign}\left(\vec{\gamma}_S^*\right)$ *holds, then* $\operatorname{sign}\left(\hat{\vec{\gamma}}\right) = \operatorname{sign}\left(\vec{\gamma}^*\right)$.

Lemma 4 (Wang et al. (2021))**.** *Assume the indenpendent zero-mean random error* $\vec{\boldsymbol{\varepsilon}}$ *is sub-Gaussian with bounded variance* $\operatorname{Var}\left(\vec{\boldsymbol{\varepsilon}}_i\right) \leq \sigma^2$*. Then with probability at least* $1 - 2cn \exp\left(-\frac{\lambda^2\eta^2}{2\sigma^2\mu_{\mathring{\boldsymbol{X}}}}\right)$ *there holds* $\left\| \mathring{\boldsymbol{X}}_{S^c}^\top \left(\boldsymbol{I} - \boldsymbol{I}_S \right) \mathring{\boldsymbol{X}} \vec{\boldsymbol{\varepsilon}} \right\|_\infty \leq \lambda\eta$ *and* $\left\| \left(\mathring{\boldsymbol{X}}_S^\top \mathring{\boldsymbol{X}}_S \right)^{-1} \mathring{\boldsymbol{X}}_S^\top \mathring{\boldsymbol{X}} \vec{\boldsymbol{\varepsilon}} \right\|_\infty \leq \frac{\lambda\eta}{\sqrt{C_{\min}\mu_{\mathring{\boldsymbol{X}}}}}$.

When we use the Puffer transformation to preconditioning Eq. (8). The assumptions of Lemma 4 is no longer satisfied while the proof of Proposition 3 do not require a independent noise, Hence it sufficient to prove a similar result of Lemma 4 with the dependent random noise.

Note that the two inequalities share the formulation of

$$\|\boldsymbol{z}\|_\infty \leq c, \quad \boldsymbol{z} = A\vec{\boldsymbol{\varepsilon}}. \tag{12}$$

And the infinity norm of $\boldsymbol{z}$ is bounded by the sum of shared upper bound for each element. Specifically, each element z_i is a weighted sum of the random noise. The point where independent noise is required by the proof technique is that a weighted sum of independent sub-Gaussian variables is still sub-Gaussian such that one can use the Hoeffding inequality to bound the weighted sum of random error.

When we face the dependent noise, we need a additional assumption that the noise is Gaussian to ensure that the weighted sum of dependent noise is still Gaussian. Then we can prove the same inequality of Eq. (12).

Lemma 5. *Assume that* $\boldsymbol{z} \in \mathbb{R}^n$ *is zero-mean Gaussian vectors. Then for any* $t > 0$*, we have*

$$\mathbb{P}(\|\boldsymbol{z}\|_\infty \geq t) \leq 2n \exp\{-\frac{t^2}{2\max_i \operatorname{Var}(z_i)}\} \tag{13}$$

Proof. We have for any $\lambda > 0$

$$\begin{aligned}
&\mathbb{E}\left[\exp\left(\lambda z_i\right)\right] \\
&= \frac{1}{\sqrt{2\pi \mathrm{Var}\left(z_i\right)}} \int_{-\infty}^{\infty} \exp\left(\lambda x\right) \exp\left(-\frac{x^2}{2\mathrm{Var}\left(z_i\right)}\right) dx \\
&= \exp\left(\frac{\lambda^2 \mathrm{Var}\left(z_i\right)}{2}\right) \frac{1}{\sqrt{2\pi \mathrm{Var}\left(z_i\right)}} \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}\left(\frac{x}{\sqrt{\mathrm{Var}\left(z_i\right)}} - \lambda\sqrt{\mathrm{Var}\left(z_i\right)}\right)^2\right) dx \\
&= \exp\left(\frac{\lambda^2 \mathrm{Var}\left(z_i\right)}{2}\right)
\end{aligned}$$

Hence

$$\begin{aligned}
\mathbb{P}\left(\|\boldsymbol{z}\|_\infty \geq t\right) &\leq \sum_i \mathbb{P}\left(|z_i| \geq t\right) \\
&= 2\sum_i \mathbb{P}\left(z_i \geq t\right) \\
&\leq 2\sum_i \inf_\lambda \exp\left(-\lambda t\right) \mathbb{E}\left[\exp\left(\lambda z_i\right)\right] \\
&= 2\sum_i \inf_\lambda \exp\left(-\lambda t\right) \exp\left(\frac{\lambda^2 \mathrm{Var}\left(z_i\right)}{2}\right) \\
&= 2\sum_i \exp\left(\frac{-t^2}{2\mathrm{Var}\left(z_i\right)}\right) \\
&\leq 2n \exp\left(\frac{-t^2}{2\max_i \mathrm{Var}\left(z_i\right)}\right)
\end{aligned}$$

□

Then we can use the same technique to proof Lemma 4 in the dependent noise setting and finish the proof of Theorem 2.