Journal of Artificial Intelligence Research 70 (2021) 1373-1411 Submitted 09/2020; published 04/2021

Confident Learning: Estimating Uncertainty in Dataset Labels

Curtis G. Northcutt cgn@mit.edu
Massachusetts Institute of Technology,
Department of EECS, Cambridge, MA, USA

Lu Jiang lujiang@google.com
Google Research, Mountain View, CA, USA

Isaac L. Chuang ichuang@mit.edu
Massachusetts Institute of Technology,
Department of EECS, Department of Physics, Cambridge, MA, USA

Abstract

Learning exists in the context of data, yet notions of *confidence* typically focus on model predictions, not label quality. Confident learning (CL) is an alternative approach which focuses instead on label quality by characterizing and identifying label errors in datasets, based on the principles of pruning noisy data, counting with probabilistic thresholds to estimate noise, and ranking examples to train with confidence. Whereas numerous studies have developed these principles independently, here, we combine them, building on the assumption of a class-conditional noise process to directly estimate the joint distribution between noisy (given) labels and uncorrupted (unknown) labels. This results in a generalized CL which is provably consistent and experimentally performant. We present sufficient conditions where CL exactly finds label errors, and show CL performance exceeding seven recent competitive approaches for learning with noisy labels on the CIFAR dataset. Uniquely, the CL framework is *not* coupled to a specific data modality or model (e.g., we use CL to find several label errors in the presumed error-free MNIST dataset and improve sentiment classification on text data in Amazon Reviews). We also employ CL on ImageNet to quantify ontological class overlap (e.g., estimating 645 *missile* images are mislabeled as their parent class *projectile*), and moderately increase model accuracy (e.g., for ResNet) by cleaning data prior to training. These results are replicable using the open-source `cleanlab` release.

1. Introduction

Advances in learning with noisy labels and weak supervision usually introduce a new model or loss function. Often this model-centric approach band-aids the real question: which data is mislabeled? Yet, large datasets with noisy labels have become increasingly common. Examples span prominent benchmark datasets like ImageNet (Russakovsky et al., 2015) and MS-COCO (Lin et al., 2014) to human-centric datasets like electronic health records (Halpern et al., 2016) and educational data (Northcutt et al., 2016). The presence of noisy labels in these datasets introduces two problems. How can we identify examples with label errors and how can we learn well despite noisy labels, irrespective of the data modality or model employed? Here, we follow a data-centric approach to theoretically and experimentally investigate the premise that the key to learning with noisy labels lies in accurately and directly characterizing the uncertainty of label noise in the data.

A large body of work, which may be termed "confident learning," has arisen to address the uncertainty in dataset labels, from which two aspects stand out. First, Angluin and Laird's (1988) classification noise process (CNP) provides a starting assumption that label noise is class-conditional, depending only on the latent true class, not the data. While there are exceptions, this assumption is commonly used (Goldberger and Ben-Reuven, 2017; Sukhbaatar et al., 2015) because it is reasonable for many datasets. For example, in ImageNet, a *leopard* is more likely to be mislabeled *jaguar* than *bathtub*. Second, direct estimation of the joint distribution between noisy (given) labels and true (unknown) labels (see Fig. 1) can be pursued effectively based on three principled approaches used in many related studies: (a) **Prune**, to search for label errors, e.g. following the example of Chen et al. (2019); Patrini et al. (2017); Van Rooyen et al. (2015), using *soft-pruning* via loss-reweighting, to avoid the convergence pitfalls of iterative re-labeling – (b) **Count**, to train on clean data, avoiding error-propagation in learned model weights from reweighting the loss (Natarajan et al., 2017) with imperfect predicted probabilities, generalizing seminal work Forman (2005, 2008); Lipton et al. (2018) – and (c) **Rank** which examples to use during training, to allow learning with unnormalized probabilities or decision boundary distances, building on well-known robustness findings (Page et al., 1997) and ideas of curriculum learning (Jiang et al., 2018).

To our knowledge, no prior work has thoroughly analyzed the direct estimation of the joint distribution between noisy and uncorrupted labels. Here, we assemble these principled approaches to generalize confident learning (CL) for this purpose. Estimating the joint distribution is challenging as it requires disambiguation of epistemic uncertainty (model predicted probabilities) from aleatoric uncertainty (noisy labels) (Chowdhary and Dupuis, 2013), but useful because its marginals yield important statistics used in the literature, including latent noise transition rates (Sukhbaatar et al., 2015; Reed et al., 2015), latent prior of uncorrupted labels (Lawrence and Schölkopf, 2001; Graepel and Herbrich, 2001), and inverse noise rates (Katz-Samuels et al., 2019). While noise rates are useful for loss-reweighting (Natarajan et al., 2013), only the joint can directly estimate the number of label errors for each pair of true and noisy classes. Removal of these errors prior to training is an effective approach for learning with noisy labels (Chen et al., 2019). The joint is also useful to discover ontological issues in datasets for dataset curation, e.g. ImageNet includes two classes for the same *maillot* class (c.f. Table 5 in Sec. 5).

The generalized CL assembled in this paper upon the principles of pruning, counting, and ranking, is a model-agnostic family of theories and algorithms for characterizing, finding, and learning with label errors. It uses predicted probabilities and noisy labels to count examples in the unnormalized *confident joint*, estimate the joint distribution, and prune noisy data, producing clean data as output.

This paper makes two key contributions to prior work on finding, understanding, and learning with noisy labels. First, a proof is presented giving realistic sufficient conditions under which CL exactly finds label errors and exactly estimates the joint distribution of noisy and true labels. Second, experimental data are shared, showing that this CL algorithm is empirically performant on three tasks (a) label noise estimation, (b) label error finding, and (c) learning with noisy labels, increasing ResNet accuracy on a cleaned-ImageNet and outperforming seven recent highly competitive methods for learning with noisy labels on the CIFAR dataset. The results presented are reproducible with the implementation of CL algorithms, open-sourced as the `cleanlab`[1] Python package.

These contributions are presented beginning with the formal problem specification and notation (Section 2), then defining the algorithmic methods employed for CL (Section 3) and theoretically bounding expected behavior under ideal and noisy conditions (Section 4). Experimental benchmarks on the CIFAR, ImageNet, WebVision, and MNIST datasets, cross-comparing CL performance with that from a wide range of highly competitive approaches, including *INCV* (Chen et al., 2019), *Mixup* (Zhang et al., 2018), *MentorNet* (Jiang et al., 2018), and *Co-Teaching* (Han et al., 2018), are then presented in Section 5. Related work (Section 6) and concluding observations (Section 7) wrap up the presentation. Extended proofs of the main theorems, algorithm details, and comprehensive performance comparison data are presented in the appendices.

2. CL Framework and Problem Set-up

In the context of multiclass data with possibly noisy labels, let $[m]$ denote $\{1, 2, ..., m\}$, the set of m unique class labels, and $\boldsymbol{X} := (\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ denote the dataset of n examples $\boldsymbol{x} \in \mathbb{R}^d$ with associated observed noisy labels $\tilde{y} \in [m]$. $\boldsymbol{x}$ and $\tilde{y}$ are coupled in $\boldsymbol{X}$ to signify that *cleaning* removes data and label. While a number of relevant works address the setting where annotator labels are available (Sambasivan et al., 2021; Bouguelia et al., 2018; Tanno et al., 2019a,b; Khetan et al., 2018), this paper addresses the general setting where no annotation information is available except the observed noisy labels.

Assumptions We assume there exists, for every example, a latent, true label y^*. Prior to observing $\tilde{y}$, a class-conditional classification noise process (Angluin and Laird, 1988) maps $y^* \to \tilde{y}$ such that every label in class $j \in [m]$ may be independently mislabeled as class $i \in [m]$ with probability $p(\tilde{y}{=}i|y^*{=}j)$. This assumption is reasonable and has been used in prior work (Goldberger and Ben-Reuven, 2017; Sukhbaatar et al., 2015).

Notation Notation is summarized in Table 1. The discrete random variable $\tilde{y}$ takes an observed, noisy label (potentially flipped to an incorrect class), and y^* takes a latent, uncorrupted label. The subset of examples in $\boldsymbol{X}$ with noisy class label i is denoted $\boldsymbol{X}_{\tilde{y}=i}$, *i.e.* $\boldsymbol{X}_{\tilde{y}=\text{cow}}$ is read, "examples with class label *cow*." The notation $p(\tilde{y}; \boldsymbol{x})$, as opposed to $p(\tilde{y}|\boldsymbol{x})$, expresses our assumption that input $\boldsymbol{x}$ is observed and error-free. We denote the discrete joint probability of the noisy and latent labels as $p(\tilde{y}, y^*)$, where conditionals $p(\tilde{y}|y^*)$ and $p(y^*|\tilde{y})$ denote probabilities of label flipping. We use $\hat{p}$ for predicted probabilities. In matrix notation, the $n \times m$ matrix of out-of-sample predicted probabilities is $\hat{\boldsymbol{P}}_{k,i} := \hat{p}(\tilde{y} = i; \boldsymbol{x}_k, \boldsymbol{\theta})$, the prior of the latent labels is $\boldsymbol{Q}_{y^*} := p(y^*{=}i)$; the $m \times m$ joint distribution matrix is $\boldsymbol{Q}_{\tilde{y},y^*} := p(\tilde{y}{=}i, y^*{=}j)$; the $m \times m$ noise transition matrix (noisy channel) of flipping rates is $\boldsymbol{Q}_{\tilde{y}|y^*} := p(\tilde{y}{=}i|y^*{=}j)$; and the $m \times m$ mixing matrix is $\boldsymbol{Q}_{y^*|\tilde{y}} := p(y^*{=}i|\tilde{y}{=}j)$. At times, we abbreviate $\hat{p}(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta})$ as $\hat{p}_{\boldsymbol{x},\tilde{y}=i}$, where $\boldsymbol{\theta}$ denotes the model parameters. CL assumes no specific loss function associated with $\boldsymbol{\theta}$: the CL framework is model-agnostic.

[1] To foster future research in data cleaning and learning with noisy labels and to improve accessibility for newcomers, `cleanlab` is open-source and well-documented: https://github.com/cgnorthcutt/cleanlab/

Table 1: Notation used in confident learning.

Notation	**Definition**
m	The number of unique class labels
$[m]$	The set of m unique class labels
$\tilde{y}$	Discrete random variable $\tilde{y} \in [m]$ takes an observed, noisy label
y^*	Discrete random variable $y^* \in [m]$ takes the unknown, true, uncorrupted label
$\boldsymbol{X}$	The dataset $(\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ of n examples $\boldsymbol{x} \in \mathbb{R}^d$ with noisy labels
$\boldsymbol{x}_k$	The k^{th} training data example
$\tilde{y}_k$	The observed, noisy label corresponding to $\boldsymbol{x}_k$
y^*_k	The unknown, true label corresponding to $\boldsymbol{x}_k$
n	The cardinality of $\boldsymbol{X} := (\boldsymbol{x}, \tilde{y})^n$, i.e. the number of examples in the dataset
$\boldsymbol{\theta}$	Model parameters
$\boldsymbol{X}_{\tilde{y}=i}$	Subset of examples in $\boldsymbol{X}$ with noisy label i, *i.e.* $\boldsymbol{X}_{\tilde{y}=\text{cat}}$ is "examples labeled cat"
$\boldsymbol{X}_{\tilde{y}=i,y^*=j}$	Subset of examples in $\boldsymbol{X}$ with noisy label i and true label j
$\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$	Estimate of subset of examples in $\boldsymbol{X}$ with noisy label i and true label j
$p(\tilde{y}{=}i, y^*{=}j)$	Discrete joint probability of noisy label i and true label j.
$p(\tilde{y}{=}i \mid y^*{=}j)$	Discrete conditional probability of true label flipping, called the noise rate
$p(y^*{=}j \mid \tilde{y}{=}i)$	Discrete conditional probability of noisy label flipping, called the inverse noise rate
$\hat{p}(\cdot)$	Estimated or predicted probability (may replace $p(\cdot)$ in any context)
$\boldsymbol{Q}_{y^*}$	The prior of the latent labels
$\hat{\boldsymbol{Q}}_{y^*}$	Estimate of the prior of the latent labels
$\boldsymbol{Q}_{\tilde{y},y^*}$	The $m \times m$ joint distribution matrix for $p(\tilde{y}, y^*)$
$\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$	Estimate of the $m \times m$ joint distribution matrix for $p(\tilde{y}, y^*)$
$\boldsymbol{Q}_{\tilde{y} \mid y^*}$	The $m \times m$ noise transition matrix (noisy channel) of flipping rates for $p(\tilde{y} \mid y^*)$
$\hat{\boldsymbol{Q}}_{\tilde{y} \mid y^*}$	Estimate of the $m \times m$ noise transition matrix of flipping rates for $p(\tilde{y} \mid y^*)$
$\boldsymbol{Q}_{y^* \mid \tilde{y}}$	The inverse noise matrix for $p(y^* \mid \tilde{y})$
$\hat{\boldsymbol{Q}}_{y^* \mid \tilde{y}}$	Estimate of the inverse noise matrix for $p(y^* \mid \tilde{y})$
$\hat{p}(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta})$	Predicted probability of label $\tilde{y} = i$ for example $\boldsymbol{x}$ and model parameters $\boldsymbol{\theta}$
$\hat{p}_{\boldsymbol{x},\tilde{y}=i}$	Shorthand abbreviation for predicted probability $\hat{p}(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta})$
$\hat{p}(\tilde{y}{=}i; \boldsymbol{x} {\in} \boldsymbol{X}_{\tilde{y}=i}, \boldsymbol{\theta})$	The *self-confidence* of example $\boldsymbol{x}$ belonging to its given label $\tilde{y}{=}i$
$\hat{\boldsymbol{P}}_{k,i}$	$n \times m$ matrix of out-of-sample predicted probabilities $\hat{p}(\tilde{y} = i; \boldsymbol{x}_k, \boldsymbol{\theta})$
$\boldsymbol{C}_{\tilde{y},y^*}$	The *confident joint* $\boldsymbol{C}_{\tilde{y},y^*} \in {\mathbb{N}_{\geq 0}}^{m \times m}$, an unnormalized estimate of $\boldsymbol{Q}_{\tilde{y},y^*}$
$\boldsymbol{C}_{\text{confusion}}$	Confusion matrix of given labels $\tilde{y}_k$ and predictions $\arg\max_{i \in [m]} \hat{p}(\tilde{y}{=}i; \boldsymbol{x}_k, \boldsymbol{\theta})$
t_j	The expected (average) self-confidence for class j used as a threshold in $\boldsymbol{C}_{\tilde{y},y^*}$
$p^*(\tilde{y}{=}i \mid y^*{=}y^*_k)$	*Ideal* probability for some example $\boldsymbol{x}_k$, equivalent to noise rate $p^*(\tilde{y}{=}i \mid y^*{=}j)$
$p^*_{\boldsymbol{x},\tilde{y}=i}$	Shorthand abbreviation for ideal probability $p^*(\tilde{y}{=}i \mid y^*{=}y^*_k)$

Goal Our assumption of a class-conditional noise process implies the label noise transitions are data-independent, i.e., $p(\tilde{y}|y^*; \boldsymbol{x}) = p(\tilde{y}|y^*)$. To characterize class-conditional label uncertainty, one must estimate $p(\tilde{y}|y^*)$ and $p(y^*)$, the latent prior distribution of uncorrupted labels. Unlike prior works which estimate $p(\tilde{y}|y^*)$ and $p(y^*)$ independently, we estimate both jointly by directly estimating the joint distribution of label noise, $p(\tilde{y}, y^*)$. **Our goal** is to estimate every $p(\tilde{y}, y^*)$ as a matrix $\boldsymbol{Q}_{\tilde{y},y^*}$ and use $\boldsymbol{Q}_{\tilde{y},y^*}$ to find all mislabeled examples $\boldsymbol{x}$ in dataset $\boldsymbol{X}$ where $y^* \neq \tilde{y}$. This is hard because it requires disambiguation of model error (epistemic uncertainty) from the intrinsic label noise (aleatoric uncertainty), while simultaneously estimating the joint distribution of label noise ($\boldsymbol{Q}_{\tilde{y},y^*}$) without prior knowledge of the latent noise transition matrix ($\boldsymbol{Q}_{\tilde{y}|y^*}$), the latent prior distribution of true labels ($\boldsymbol{Q}_{y^*}$), or any latent, true labels ($y*$).

Definition 1 (Sparsity)**.** *A statistic to quantify the characteristic shape of the label noise defined by fraction of zeros in the off-diagonals of* $\boldsymbol{Q}_{\tilde{y},y^*}$. High sparsity quantifies non-uniformity of label noise, common to real-world datasets. For example, in ImageNet, *missile* may have high probability of being mislabeled as *projectile*, but near-zero probability of being mislabeled as most other classes like *wool* or *wine*. Zero sparsity implies every noise rate in $\boldsymbol{Q}_{\tilde{y},y^*}$ is non-zero. A sparsity of 1 implies no label noise because the off-diagonals of $\boldsymbol{Q}_{\tilde{y},y^*}$, which encapsulate the class-conditional noise rates, must all be zero if sparsity $= 1$.

Definition 2 (Self-Confidence)**.** *The predicted probability for some model* $\boldsymbol{\theta}$ *that an example* $\boldsymbol{x}$ *belongs to its given label* $\tilde{y}$*, expressed as* $\hat{p}(\tilde{y}{=}i; \boldsymbol{x}{\in}\boldsymbol{X}_{\tilde{y}=i}, \boldsymbol{\theta})$. Low self-confidence is a heuristic-likelihood of being a label error.

3. CL Methods

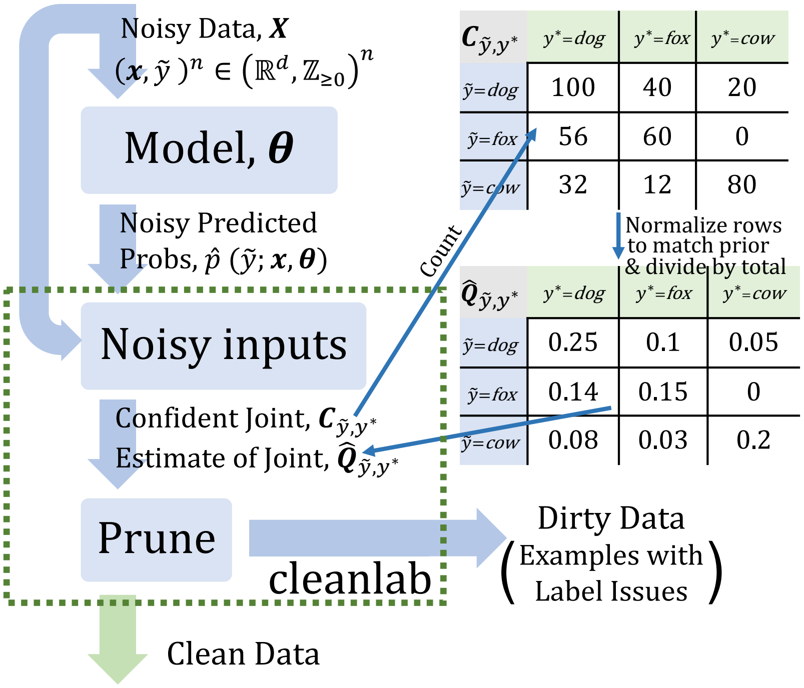
Confident learning (CL) estimates the joint distribution between the (noisy) observed labels and the (true) latent labels. CL requires two inputs: (1) the out-of-sample predicted probabilities $\hat{\boldsymbol{P}}_{k,i}$ and (2) the vector of noisy labels $\tilde{y}_k$. The two inputs are linked via index k for all $\boldsymbol{x}_k \in \boldsymbol{X}$. None of the true labels y^* are available, except when $\tilde{y} = y^*$, and we do not know when that is the case.

The out-of-sample predicted probabilities $\hat{\boldsymbol{P}}_{k,i}$ used as input to CL are computed beforehand (e.g. cross-validation) using a model $\boldsymbol{\theta}$: so, how does $\boldsymbol{\theta}$ fit into the CL framework? Prior works typically learn with noisy labels by directly modifying the model or training loss function, restricting the class of models. Instead, CL decouples the model and data cleaning procedure by working with model outputs $\hat{\boldsymbol{P}}_{k,i}$, so that any model that produces a mapping $\boldsymbol{\theta} : \boldsymbol{x} \to \hat{p}(\tilde{y}{=}i; \boldsymbol{x}_k, \boldsymbol{\theta})$ can be used (e.g. neural nets with a softmax output, naive Bayes, logistic regression, etc.). However, $\boldsymbol{\theta}$ affects the predicted probabilities $\hat{p}(\tilde{y}{=}i; \boldsymbol{x}_k, \boldsymbol{\theta})$ which in turn affect the performance of CL. Hence, in Section 4, we examine sufficient conditions where CL finds label errors exactly, even when $\hat{p}(\tilde{y}{=}i; \boldsymbol{x}_k, \boldsymbol{\theta})$ is erroneous. Any model $\boldsymbol{\theta}$ may be used for final training on clean data provided by CL.

CL identifies noisy labels in existing datasets to improve learning with noisy labels. The main procedure (see Fig. 1) comprises three steps: (1) estimate $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ to characterize class-conditional label noise (Sec. 3.1), (2) filter out noisy examples (Sec. 3.2), and (3) train with errors removed, reweighting the examples by class weights $\frac{\hat{\boldsymbol{Q}}_{y^*}[i]}{\hat{\boldsymbol{Q}}_{\tilde{y},y^*}[i][i]}$ for each class $i \in [m]$. In this section, we define these three steps and discuss their expected outcomes.

3.1 Count: Characterize and Find Label Errors using the Confident Joint

To estimate the joint distribution of noisy labels $\tilde{y}$ and true labels, $\boldsymbol{Q}_{\tilde{y},y^*}$, we count examples that are likely to belong to another class and calibrate those counts so that they sum to the given count of noisy labels in each class, $|\boldsymbol{X}_{\tilde{y}=i}|$. Counts are captured in the *confident joint* $\boldsymbol{C}_{\tilde{y},y^*} \in {\mathbb{Z}_{\geq 0}}^{m \times m}$, a statistical data structure in CL to directly find label errors. Diagonal entries of $\boldsymbol{C}_{\tilde{y},y^*}$ count correct labels and non-diagonals capture asymmetric label error counts. As an example, $C_{\tilde{y}=3,y^*=1}{=}10$ is read, “Ten examples are labeled *3* but should be labeled *1*.”


Noisy Data, $\boldsymbol{X}$
$(\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, \mathbb{Z}_{\geq 0})^n$
Model, $\boldsymbol{\theta}$
Noisy Predicted
Probs, $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$
Noisy inputs
Confident Joint, $\boldsymbol{C}_{\tilde{y},y^*}$
Estimate of Joint, $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$
Prune
cleanlab
Clean Data
Dirty Data
(Examples with
Label Issues)
Count
Normalize rows
to match prior
& divide by total

$\boldsymbol{C}_{\tilde{y},y^*}$	y^*=dog	y^*=fox	y^*=cow
$\tilde{y}$=dog	100	40	20
$\tilde{y}$=fox	56	60	0
$\tilde{y}$=cow	32	12	80

$\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$	y^*=dog	y^*=fox	y^*=cow
$\tilde{y}$=dog	0.25	0.1	0.05
$\tilde{y}$=fox	0.14	0.15	0
$\tilde{y}$=cow	0.08	0.03	0.2

Figure 1: An example of the confident learning (CL) process. CL uses the confident joint, $\boldsymbol{C}_{\tilde{y},y^*}$, and $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$, an estimate of $\boldsymbol{Q}_{\tilde{y},y^*}$, the joint distribution of noisy observed labels $\tilde{y}$ and unknown true labels y^*, to find examples with label errors and produce clean data for training.

In this section, we first introduce the *confident joint* $\boldsymbol{C}_{\tilde{y},y^*}$ to partition and count label errors. Second, we show how $\boldsymbol{C}_{\tilde{y},y^*}$ is used to estimate $\boldsymbol{Q}_{\tilde{y},y^*}$ and characterize label noise in a dataset $\boldsymbol{X}$. Finally, we provide a related baseline $\boldsymbol{C}_{\text{confusion}}$ and consider its assumptions and short-comings (e.g. class-imbalance) in comparison with $\boldsymbol{C}_{\tilde{y},y^*}$ and CL. CL overcomes these shortcomings using thresholding and collision handling to enable robustness to class imbalance and heterogeneity in predicted probability distributions across classes.

The confident joint $\boldsymbol{C}_{\tilde{y},y^*}$ $\boldsymbol{C}_{\tilde{y},y^*}$ estimates $\boldsymbol{X}_{\tilde{y}=i,y^*=j}$, the set of examples with noisy label i that actually have true label j, by partitioning $\boldsymbol{X}$ into estimate bins $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$. When $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$, then $\boldsymbol{C}_{\tilde{y},y^*}$ exactly finds label errors (proof in Sec. 4). $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ (note the hat above $\hat{\boldsymbol{X}}$ to indicate $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ is an estimate of $\boldsymbol{X}_{\tilde{y}=i,y^*=j}$) is the set of examples $\boldsymbol{x}$ labeled $\tilde{y}$=i with *large enough* $\hat{p}(\tilde{y}=j; \boldsymbol{x}, \boldsymbol{\theta})$ to likely belong to class y^*=j, determined by a per-class threshold, t_j. Formally, the definition of the *confident joint* is

$$\begin{aligned} &\boldsymbol{C}_{\tilde{y},y^*}[i][j] := |\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}| \quad \text{where} \\ &\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} := \left\{ \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \hat{p}(\tilde{y}=j; \boldsymbol{x}, \boldsymbol{\theta}) \geq t_j, \; j = \operatorname*{arg\,max}_{l \in [m] : \hat{p}(\tilde{y}=l; \boldsymbol{x}, \boldsymbol{\theta}) \geq t_l} \hat{p}(\tilde{y}=l; \boldsymbol{x}, \boldsymbol{\theta}) \right\} \end{aligned} \tag{1}$$

and the threshold t_j is the expected (average) self-confidence for each class

$$t_j = \frac{1}{|\boldsymbol{X}_{\tilde{y}=j}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \hat{p}(\tilde{y}=j; \boldsymbol{x}, \boldsymbol{\theta}) \tag{2}$$

Unlike prior art, which estimates label errors under the assumption that the true labels are $\tilde{y}^*_k = \arg\max_{i \in [m]} \hat{p}(\tilde{y}{=}i; \boldsymbol{x}_k, \boldsymbol{\theta})$ (Chen et al., 2019), the thresholds in this formulation improve CL uncertainty quantification robustness to (1) heterogeneous class probability distributions and (2) class-imbalance. For example, if examples labeled i tend to have higher probabilities because the model is over-confident about class i, then t_i will be proportionally larger; if some other class j tends toward low probabilities, t_j will be smaller. These thresholds allow us to guess y^* in spite of class-imbalance, unlike prior art which may guess over-confident classes for y^* because $\arg\max$ is used (Guo et al., 2017). We examine "how good" the probabilities produced by model $\boldsymbol{\theta}$ need to be for this approach to work in Section 4.

To disentangle Eqn. 1, consider a simplified formulation:

$$\hat{\boldsymbol{X}}^{(\text{simple})}_{\tilde{y}=i, y^*=j} = \{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \; \hat{p}(\tilde{y}=j; \boldsymbol{x}, \boldsymbol{\theta}) \geq t_j\}$$

The simplified formulation, however, introduces *label collisions* when an example $\boldsymbol{x}$ is confidently counted into more than one $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ bin. Collisions only occur along the y^* dimension of $\boldsymbol{C}_{\tilde{y},y^*}$ because $\tilde{y}$ is given. We handle collisions in the right-hand side of Eqn. 1 by selecting $\hat{y}^* \leftarrow \arg\max_{j \in [m]} \; \hat{p}(\tilde{y}=j; \boldsymbol{x}, \boldsymbol{\theta})$ whenever $|\{k{\in}[m] : \hat{p}(\tilde{y}{=}k; \boldsymbol{x}{\in}\boldsymbol{X}_{\tilde{y}=i}, \boldsymbol{\theta}) \geq t_k\}| > 1$ (collision). In practice with softmax, collisions sometimes occur for softmax outputs with higher temperature (more uniform probabilities), few collisions occur with lower temperature, and no collisions occur with a temperature of zero (one-hot prediction probabilities).

The definition of $\boldsymbol{C}_{\tilde{y},y^*}$ in Eqn. 1 has some nice properties in certain circumstances. First, if an example has low (near-uniform) predicted probabilities across classes, then it will not be counted for any class in $\boldsymbol{C}_{\tilde{y},y^*}$ so that $\boldsymbol{C}_{\tilde{y},y^*}$ may be robust to pure noise or examples from an alien class not in the dataset. Second, $\boldsymbol{C}_{\tilde{y},y^*}$ is intuitive – t_j embodies the intuition that examples with higher probability of belonging to class j than the expected probability of examples in class j probably belong to class j. Third, thresholding allows flexibility – for example, the 90^{th} percentile may be used in t_j instead of the mean to find errors with higher confidence; despite the flexibility, we use the mean because we show (in Sec. 4) that this formulation exactly finds label errors in various settings, and we leave the study of other formulations, like a percentile-based threshold, as future work.

Complexity We provide algorithmic implementations of Eqns. 2, 1, and 3 in the Appendix. Given predicted probabilities $\hat{\boldsymbol{P}}_{k,i}$ and noisy labels $\tilde{y}$, these require $\mathcal{O}(m^2 + nm)$ storage and arithmetic operations to compute $\boldsymbol{C}_{\tilde{y},y^*}$, for n training examples over m classes.

Estimate the joint $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$. Given the confident joint $\boldsymbol{C}_{\tilde{y},y^*}$, we estimate $\boldsymbol{Q}_{\tilde{y},y^*}$ as

$$\hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j} = \frac{\frac{\boldsymbol{C}_{\tilde{y}=i,y^*=j}}{\sum_{j \in [m]} \boldsymbol{C}_{\tilde{y}=i,y^*=j}} \cdot |\boldsymbol{X}_{\tilde{y}=i}|}{\sum\limits_{i \in [m], j \in [m]} \left(\frac{\boldsymbol{C}_{\tilde{y}=i,y^*=j}}{\sum_{j' \in [m]} \boldsymbol{C}_{\tilde{y}=i,y^*=j'}} \cdot |\boldsymbol{X}_{\tilde{y}=i}| \right)} \tag{3}$$

The numerator calibrates $\sum_j \hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j} = |\boldsymbol{X}_i| / \sum_{i \in [m]} |\boldsymbol{X}_i|, \forall i{\in}[m]$ so that row-sums match the observed marginals. The denominator calibrates $\sum_{i,j} \hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j} = 1$ so that the distribution sums to 1.

Label noise characterization Using the observed prior $\boldsymbol{Q}_{\tilde{y}=i} = |\boldsymbol{X}_i| \,/\, \sum_{i\in[m]}|\boldsymbol{X}_i|$ and marginals of $\boldsymbol{Q}_{\tilde{y},y^*}$, we estimate the latent prior as $\hat{\boldsymbol{Q}}_{y^*=j} \coloneqq \sum_i \hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j}, \forall j{\in}[m]$; the noise transition matrix (noisy channel) as $\hat{\boldsymbol{Q}}_{\tilde{y}=i|y^*=j} \coloneqq \hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j}/\hat{\boldsymbol{Q}}_{y^*=j}, \forall i{\in}[m]$; and the mixing matrix (Katz-Samuels et al., 2019) as $\hat{\boldsymbol{Q}}_{y^*=j|\tilde{y}=i} \coloneqq \hat{\boldsymbol{Q}}^{\top}_{\tilde{y}=j,y^*=i}/\boldsymbol{Q}_{\tilde{y}=i}, \forall i{\in}[m]$. As long as $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$, each of these estimators is similarly consistent (we prove this is the case under practical conditions in Sec. 4). Whereas prior approaches compute the noise transition matrices by directly averaging error-prone predicted probabilities (Reed et al., 2015; Goldberger and Ben-Reuven, 2017), CL is one step removed from the predicted probabilities by estimating noise rates based on counts from $\boldsymbol{C}_{\tilde{y},y^*}$ – these counts are computed based on whether the predicted probability is greater than a threshold, relying only on the *relative ranking* of the predicted probability, not its exact value. This feature lends itself to the robustness of confident learning to imperfect probability estimation.

Baseline approach $\boldsymbol{C}_{\text{confusion}}$ To situate our understanding of $\boldsymbol{C}_{\tilde{y},y^*}$ performance in the context of prior work, we compare $\boldsymbol{C}_{\tilde{y},y^*}$ with $\boldsymbol{C}_{\text{confusion}}$, a baseline based on a single-iteration of the performant INCV method (Chen et al., 2019). $\boldsymbol{C}_{\text{confusion}}$ forms an $m \times m$ confusion matrix of counts $|\tilde{y}_k = i, y^*_k = j|$ across all examples $\boldsymbol{x}_k$, assuming that model predictions, trained from noisy labels, uncover the true labels, i.e. $\boldsymbol{C}_{\text{confusion}}$ simply assumes $y^*_k = \arg\max_{i\in[m]} \hat{p}(\tilde{y}{=}i; \boldsymbol{x}_k, \boldsymbol{\theta})$. This baseline approach performs reasonably empirically (Sec. 5) and is a consistent estimator for noiseless predicted probabilities (Thm. 1), but fails when the distributions of probabilities are not similar for each class (Thm. 2), e.g. class-imbalance, or when predicted probabilities are overconfident (Guo et al., 2017).

Comparison of $\boldsymbol{C}_{\tilde{y},y^*}$ (confident joint) with $\boldsymbol{C}_{\text{confusion}}$ (baseline) To overcome the sensitivity of $\boldsymbol{C}_{\text{confusion}}$ to class-imbalance and distribution heterogeneity, the *confident joint*, $\boldsymbol{C}_{\tilde{y},y^*}$, uses per-class thresholding (Richard and Lippmann, 1991; Elkan, 2001) as a form of calibration (Hendrycks and Gimpel, 2017). Moreover, we prove that unlike $\boldsymbol{C}_{\text{confusion}}$, the confident joint (Eqn. 1) exactly finds label errors and consistently estimates $\boldsymbol{Q}_{\tilde{y},y^*}$ in more realistic settings with noisy predicted probabilities (see Sec. 4, Thm. 2).

3.2 Rank and Prune: Data Cleaning

Following the estimation of $\boldsymbol{C}_{\tilde{y},y^*}$ and $\boldsymbol{Q}_{\tilde{y},y^*}$ (Section 3.1), any rank and prune approach can be used to clean data. This *modularity* property allows CL to find label errors using interpretable and explainable ranking methods, whereas prior works typically couple estimation of the noise transition matrix with training loss (Goldberger and Ben-Reuven, 2017) or couple the label confidence of each example with the training loss using loss reweighting (Natarajan et al., 2013; Jiang et al., 2018). In this paper, we investigate and evaluate five rank and prune methods for finding label errors, grouped into two approaches. We provide a theoretical analysis for Method 2: $\boldsymbol{C}_{\tilde{y},y^*}$ in Sec. 4 and evaluate all methods empirically in Sec. 5.

Approach 1: Use off-diagonals of $\boldsymbol{C}_{\tilde{y},y^*}$ to estimate $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ We directly use the sets of examples counted in the off-diagonals of $\boldsymbol{C}_{\tilde{y},y^*}$ to estimate label errors.

CL baseline 1: $\boldsymbol{C}_{confusion}$. Estimate label errors as the Boolean vector $\tilde{y}_k \neq \arg\max_{j\in[m]} \hat{p}(\tilde{y}=j; \boldsymbol{x}_k, \boldsymbol{\theta})$, for all $\boldsymbol{x}_k{\in}\boldsymbol{X}$, where *true* implies label error and *false* implies clean data. This is identical to using the off-diagonals of $\boldsymbol{C}_{\text{confusion}}$ and similar to a single iteration of INCV (Chen et al., 2019).

CL method 2: $\boldsymbol{C}_{\tilde{y},y^*}$. Estimate label errors as $\{\boldsymbol{x} \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} : i \neq j\}$ from the off-diagonals of $\boldsymbol{C}_{\tilde{y},y^*}$.

Approach 2: Use $n \cdot \hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ to estimate $|\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}|$, prune by probability ranking These approaches calculate $n \cdot \hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ to estimate $|\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}|$, the count of label errors in each partition. They either sum over the y^* dimension of $|\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}|$ to estimate and remove the number of errors in each class (prune by class), or prune for every off-diagonal partition (*prune by noise rate*). The choice of which examples to remove is made by ranking the examples based on predicted probabilities.

CL method 3: Prune by Class (PBC). For each class $i \in [m]$, select the $n \cdot \sum_{j \in [m]: j \neq i} \left(\hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j}[i] \right)$ examples with lowest self-confidence $\hat{p}(\tilde{y}=i; \boldsymbol{x} \in \boldsymbol{X}_i)$.

CL method 4: Prune by Noise Rate (PBNR). For each off-diagonal entry in $\hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j}, i \neq j$, select the $n \cdot \hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j}$ examples $\boldsymbol{x} {\in} \boldsymbol{X}_{\tilde{y}=i}$ with max margin $\hat{p}_{\boldsymbol{x},\tilde{y}=j} - \hat{p}_{\boldsymbol{x},\tilde{y}=i}$. This margin is adapted from Wei et al.'s (2018) normalized margin.

CL method 5: C+NR. Combine the previous two methods via element-wise '*and*', i.e. set intersection. Prune an example if both methods PBC and PBNR prune that example.

Learning with Noisy Labels To train with errors removed, we account for missing data by reweighting the loss by $\frac{1}{\hat{p}(\tilde{y}=i|y^*=i)} {=} \frac{\hat{\boldsymbol{Q}}_{y^*}[i]}{\hat{\boldsymbol{Q}}_{\tilde{y},y^*}[i][i]}$ for each class $i {\in} [m]$, where dividing by $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}[i][i]$ normalizes out the count of clean training data and $\hat{\boldsymbol{Q}}_{y^*}[i]$ re-normalizes to the latent number of examples in class i. CL finds errors, but does not prescribe a specific training procedure using the clean data. Theoretically, CL requires no hyper-parameters to find label errors. In practice, cross-validation might introduce a hyper-parameter: k-fold. However, in our paper $k = 4$ is fixed in the experiments using cross-validation.

Which CL method to use? Five methods are presented to clean data. By default we use CL: $\boldsymbol{C}_{\tilde{y},y^*}$ because it matches the conditions of Thm. 2 exactly and is experimentally performant (see Table 4). Once label errors are found, we observe ordering label errors by the normalized margin: $\hat{p}(\tilde{y}{=}i; \boldsymbol{x}, \boldsymbol{\theta}) - \max_{j \neq i} \hat{p}(\tilde{y}{=}j; \boldsymbol{x}, \boldsymbol{\theta})$ (Wei et al., 2018) works well.

4. Theory

In this section, we examine sufficient conditions when (1) the confident joint exactly finds label errors and (2) $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ is a consistent estimator for $\boldsymbol{Q}_{\tilde{y},y^*}$. We first analyze CL for noiseless $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$, then evaluate more realistic conditions, culminating in Thm. 2 where we prove (1) and (2) with noise in the predicted probabilities for every example. Proofs are in the Appendix (see Sec. A). As a notation reminder, $\hat{p}_{\boldsymbol{x},\tilde{y}=i}$ is shorthand for $\hat{p}(\tilde{y}{=}i; \boldsymbol{x}, \boldsymbol{\theta})$.

In the statement of each theorem, we use $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$, i.e. *approximately equals*, to account for precision error of using discrete count-based $\boldsymbol{C}_{\tilde{y},y^*}$ to estimate real-valued $\boldsymbol{Q}_{\tilde{y},y^*}$. For example, if a noise rate is 0.39, but the dataset has only 5 examples in that class, the nearest possible estimate by removing errors is $2/5 = 0.4 \approxeq 0.39$. So, $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ is technically a *consistent estimator* for $\boldsymbol{Q}_{\tilde{y},y^*}$ only because of discretization error, otherwise all equalities are exact. Throughout, we assume $\boldsymbol{X}$ includes at least one example from every class.

4.1 Noiseless Predicted Probabilities

We start with the *ideal* condition and a non-obvious lemma that yields a closed-form expression for threshold t_i when $\hat{p}_{\boldsymbol{x},\tilde{y}=i}$ is ideal. Without some condition on $\hat{p}_{\boldsymbol{x},\tilde{y}=i}$, one cannot disambiguate label noise from model noise.

Condition 1 (Ideal)**.** *The predicted probabilities* $\hat{p}(\tilde{y};\boldsymbol{x},\boldsymbol{\theta})$ *for a model* θ *are* ideal *if* $\forall \boldsymbol{x}_k{\in}\boldsymbol{X}_{y^*=j}, i{\in}[m], j{\in}[m]$, *we have that* $\hat{p}(\tilde{y}{=}i;\boldsymbol{x}_k \in \boldsymbol{X}_{y^*=j},\boldsymbol{\theta}) = p^*(\tilde{y}{=}i|y^*{=}y^*_k) = p^*(\tilde{y}{=}i|y^*{=}j)$. The final equality follows from the class-conditional noise process assumption. The *ideal* condition implies error-free predicted probabilities: they match the noise rates corresponding to the y^* label of $\boldsymbol{x}$. We use $p^*_{\boldsymbol{x},\tilde{y}=i}$ as a shorthand.

Lemma 1 (Ideal Thresholds)**.** *For a noisy dataset* $\boldsymbol{X} \coloneqq (\boldsymbol{x},\tilde{y})^n \in (\mathbb{R}^d,[m])^n$ *and model* $\boldsymbol{\theta}$, *if* $\hat{p}(\tilde{y};\boldsymbol{x},\boldsymbol{\theta})$ *is* ideal, *then* $\forall i{\in}[m], t_i = \sum_{j\in[m]} p(\tilde{y}=i|y^*{=}j)p(y^*{=}j|\tilde{y}=i)$.

This form of the threshold is intuitively reasonable: the contributions to the sum when $i=j$ represents the probabilities of correct labeling, whereas when $i \neq j$, the terms give the probabilities of mislabeling $p(\tilde{y}=i|y^*=j)$, weighted by the probability $p(y^*=j|\tilde{y}=i)$ that the mislabeling is corrected. Using Lemma 1 under the ideal condition, we prove in Thm. 1 confident learning exactly finds label errors and $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ is a consistent estimator for $\boldsymbol{Q}_{\tilde{y},y^*}$ when each diagonal entry of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its row and column. The proof hinges on the fact that the construction of $\boldsymbol{C}_{\tilde{y},y^*}$ eliminates collisions.

Theorem 1 (Exact Label Errors)**.** *For a noisy dataset,* $\boldsymbol{X} \coloneqq (\boldsymbol{x},\tilde{y})^n{\in}(\mathbb{R}^d,[m])^n$ *and model* $\boldsymbol{\theta}{:}\boldsymbol{x}{\to}\hat{p}(\tilde{y})$, *if* $\hat{p}(\tilde{y};\boldsymbol{x},\boldsymbol{\theta})$ *is* ideal *and each diagonal entry of* $\boldsymbol{Q}_{\tilde{y}|y^*}$ *maximizes its row and column, then* $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ *and* $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$ *(consistent estimator for* $\boldsymbol{Q}_{\tilde{y},y^*}$*).*

While Thm. 1 is a reasonable sanity check, observe that $y^* \leftarrow \arg\max_j \; \hat{p}(\tilde{y}{=}i|\tilde{y}^*{=}i;\boldsymbol{x})$, used by $\boldsymbol{C}_{\text{confusion}}$, trivially satisfies Thm. 1 if the diagonal of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its row and column. We highlight this because $\boldsymbol{C}_{\text{confusion}}$ is the variant of CL most-related to prior work (e.g., Chen et al. (2019)). We next consider relaxed conditions *motivated by real-world settings* (e.g., Jiang et al. (2020)) where $\boldsymbol{C}_{\tilde{y},y^*}$ exactly finds label errors ($\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$) and consistently estimates the joint distribution of noisy and true labels ($\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$), but $\boldsymbol{C}_{\text{confusion}}$ does not.

4.2 Noisy Predicted Probabilities

Motivated by the importance of addressing class imbalance and heterogeneous class probability distributions, we consider linear combinations of noise per-class. Here, we index $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ by j to match the comparison $\hat{p}(\tilde{y}{=}j;\boldsymbol{x},\boldsymbol{\theta}) \geq t_j$ from the construction of $\boldsymbol{C}_{\tilde{y},y^*}$ (see Eqn. 1).

Condition 2 (Per-Class Diffracted)**.** $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ *is* per-class diffracted *if there exist linear combinations of class-conditional error in the predicted probabilities s.t.* $\hat{p}_{\boldsymbol{x},\tilde{y}=j} = \epsilon_j^{(1)} p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)}$ *where* $\epsilon_j^{(1)}, \epsilon_j^{(2)} \in \mathbb{R}$ *and* ϵ_j *can be any distribution.* This relaxes the *ideal* condition with noise that is relevant for neural networks, which are known to be class-conditionally overly confident (Guo et al., 2017).

Corollary 1.1 (Per-Class Robustness)**.** *For a noisy dataset,* $\boldsymbol{X} \coloneqq (\boldsymbol{x},\tilde{y})^n{\in}(\mathbb{R}^d,[m])^n$ *and model* $\boldsymbol{\theta}{:}\boldsymbol{x}{\to}\hat{p}(\tilde{y})$, *if* $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ *is* ***per-class diffracted*** *without label collisions and each diagonal entry of* $\boldsymbol{Q}_{\tilde{y}|y^*}$ *maximizes its row, then* $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ *and* $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$.

Cor. 1.1 shows us that $\boldsymbol{C}_{\tilde{y},y^*}$ in confident learning (which counts $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$) is robust to any linear combination of per-class error in probabilities. This is not the case for $\boldsymbol{C}_{\text{confusion}}$ because Cor. 1.1 no longer requires that the diagonal of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximize its column as before in Thm. 1: for intuition, consider an extreme case of per-class diffraction where the probabilities of only one class are all dramatically increased. Then $\boldsymbol{C}_{\text{confusion}}$, which relies on $\tilde{y}^*_k \leftarrow \arg\max_{i\in[m]} \hat{p}(\tilde{y}{=}i|y^*{=}j;\boldsymbol{x}_k)$, will count only that one class for all y^* such that all entries in the $\boldsymbol{C}_{\text{confusion}}$ will be zero except for one column, i.e. $\boldsymbol{C}_{\text{confusion}}$ cannot count entries in any other column, so $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \neq \boldsymbol{X}_{\tilde{y}=i,y^*=j}$. In comparison, for $\boldsymbol{C}_{\tilde{y},y^*}$, the increased probabilities of the one class would be subtracted by the class-threshold, re-normalizing the columns of the matrix, such that, $\boldsymbol{C}_{\tilde{y},y^*}$ satisfies Cor. 1.1 using thresholds for robustness to distributional shift and class-imbalance.

Cor. 1.1 only allows for m alterations in the probabilities and there are only m^2 unique probabilities under the ideal condition, whereas in real-world conditions, an error-prone model could potentially output $n \times m$ unique probabilities. Next, in Thm. 2, we examine a reasonable sufficient condition where CL is robust to erroneous probabilities for every example and class.

Condition 3 (Per-Example Diffracted)**.** *$\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ is* per-example diffracted *if $\forall j{\in}[m], \forall \boldsymbol{x}{\in}\boldsymbol{X}$, we have error as $\hat{p}_{\boldsymbol{x},\tilde{y}=j} = p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j}$ where*

$$\epsilon_{\boldsymbol{x},\tilde{y}=j} \sim \begin{cases} \mathcal{U}(\epsilon_j{+}t_j{-}p^*_{\boldsymbol{x},\tilde{y}=j}\,,\ \epsilon_j{-}t_j{+}p^*_{\boldsymbol{x},\tilde{y}=j}] & p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j \\ \mathcal{U}[\epsilon_j{-}t_j{+}p^*_{\boldsymbol{x},\tilde{y}=j}\,,\ \epsilon_j{+}t_j{-}p^*_{\boldsymbol{x},\tilde{y}=j}) & p^*_{\boldsymbol{x},\tilde{y}=j} < t_j \end{cases} \tag{4}$$

where $\epsilon_j = \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}}\left[\epsilon_{\boldsymbol{x},\tilde{y}=j}\right]$ and $\mathcal{U}$ denotes a uniform distribution (we discuss a more general case in the Appendix).

Theorem 2 (Per-Example Robustness)**.** *For a noisy dataset, $\boldsymbol{X} := (\boldsymbol{x},\tilde{y})^n \in (\mathbb{R}^d,[m])^n$ and model $\boldsymbol{\theta}{:}\boldsymbol{x}{\to}\hat{p}(\tilde{y})$, if $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ is* ***per-example diffracted*** *without label collisions and each diagonal entry of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its row, then $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \approxeq \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ and $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$.*

In Thm. 2, we observe that if each example's predicted probability resides within the residual range of the ideal probability and the threshold, then CL exactly identifies the label errors and consistently estimates $\boldsymbol{Q}_{\tilde{y},y^*}$. Intuitively, if $\hat{p}_{\boldsymbol{x},\tilde{y}=j} \geq t_j$ whenever $p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j$ and $\hat{p}_{\boldsymbol{x},\tilde{y}=j} < t_j$ whenever $p^*_{\boldsymbol{x},\tilde{y}=j} < t_j$, then regardless of error in $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$, CL exactly finds label errors. As an example, consider an image $\boldsymbol{x}_k$ that is mislabeled as *fox*, but is actually a *dog* where $t_{fox} = 0.6$, $p^*(\tilde{y}{=}fox;\boldsymbol{x} \in \boldsymbol{X}_{y^*=dog},\boldsymbol{\theta}) = 0.2$, $t_{dog} = 0.8$, and $p^*(\tilde{y}{=}dog;\boldsymbol{x} \in \boldsymbol{X}_{y^*=dog},\boldsymbol{\theta}) = 0.9$. Then as long as $-0.4 \leq \epsilon_{\boldsymbol{x},fox} < 0.4$ and $-0.1 < \epsilon_{\boldsymbol{x},dog} \leq 0.1$, CL will surmise $y^*_k = dog$, not fox, even though $\tilde{y}_k = fox$ is given. We empirically substantiate this theoretical result in Section 5.2.

Thm. 2 addresses the *epistemic* uncertainty of latent label noise, via the statistic, $\boldsymbol{Q}_{\tilde{y},y^*}$, while accounting for the *aleatoric* uncertainty of inherently erroneous predicted probabilities.

5. Experiments

This section empirically validates CL on CIFAR (Krizhevsky and Hinton, 2009) and ImageNet (Russakovsky et al., 2015) benchmarks. Sec. 5.1 presents CL performance on noisy examples in CIFAR where true labels are presumed known. Sec. 5.2 shows real-world label errors found in the original, unperturbed MNIST, ImageNet, WebVision, and Amazon Reviews datasets, and shows performance advantages using cleaned data provided by CL to train ImageNet. Unless otherwise specified, we compute out-of-sample predicted probabilities $\hat{\boldsymbol{P}}_{k,j}$ using four-fold cross-validation and ResNet architectures.

5.1 Asymmetric Label Noise on CIFAR-10 dataset

We evaluate CL on three criteria: (a) joint estimation (Fig. 2), (b) accuracy finding label errors (Table 4), and (c) accuracy learning with noisy labels (Table 2).

Noise Generation Following prior work by Sukhbaatar et al. (2015); Goldberger and Ben-Reuven (2017), we verify CL performance on the commonly used asymmetric label noise, where the labels of error-free/clean data are randomly flipped, for its resemblance to real-world noise. We generate noisy data from clean data by randomly switching some labels of training examples to different classes non-uniformly according to a randomly generated $\boldsymbol{Q}_{\tilde{y}|y^*}$ noise transition matrix. We generate $\boldsymbol{Q}_{\tilde{y}|y^*}$ matrices with different traces to run experiments for different noise levels. The noise matrices used in our experiments are in the Appendix in Fig. S3. We generate noise in the CIFAR-10 training dataset across varying *sparsities*, the fraction of off-diagonals in $\boldsymbol{Q}_{\tilde{y},y^*}$ that are zero, and the percent of incorrect labels (noise). We evaluate all models on the unaltered test set.

Baselines and our method In Table 2, we compare CL performance versus seven recent highly competitive approaches and a vanilla baseline for multiclass learning with noisy labels on CIFAR-10, including *INCV* (Chen et al., 2019) which finds clean data with multiple iterations of cross-validation then trains on the clean set, *SCE-loss* (symmetric cross entropy) (Wang et al., 2019) which adds a reverse cross entropy term for loss-correction, *Mixup* (Zhang et al., 2018) which linearly combines examples and labels to augment data, *MentorNet* (Jiang et al., 2018) which uses curriculum learning to avoid noisy data in training, *Co-Teaching* (Han et al., 2018) which trains two models in tandem to learn from clean data, *S-Model* (Goldberger and Ben-Reuven, 2017) which uses an extra softmax layer to model noise during training, and *Reed* (Reed et al., 2015) which uses loss-reweighting; and a *Baseline* model that denotes a vanilla training with the noisy labels.

Training settings All models are trained using ResNet-50 with the common setting: learning rate 0.1 for epoch [0,150), 0.01 for epoch [150,250), 0.001 for epoch [250,350); momentum 0.9; and weight decay 0.0001, except *INCV*, *SCE-loss*, and *Co-Teaching* which are trained using their official GitHub code. Settings are copied from the kuangliu/pytorch-cifar GitHub open-source code and were not tuned by hand. We report the highest score across hyper-parameters $\alpha \in \{1, 2, 4, 8\}$ for *Mixup* and $p \in \{0.7, 0.8, 0.9\}$ for *MentorNet*. For fair comparison with *Co-Teaching*, *INCV*, and *MentorNet*, we also train using the *co-teaching* approach with forget rate $= 0.5 \times$ [noise fraction], and report the max accuracy of the two trained models for each method. We observe that dropping the last partial batch of each epoch during training improves stability by avoiding weight updates from, in some cases, a single noisy example). Exactly the same noisy labels are used for training all models for each column of Table 2. For our method, we fix its hyper-parameter, *i.e.* the number of folds in cross-validation across different noise levels, and do not tune it on the validation set.

Table 2: Test accuracy (%) of confident learning versus recent methods for learning with noisy labels in CIFAR-10. Scores reported for CL methods are averaged over ten trials with standard deviations shown in Table 3. CL methods estimate label errors, remove them, then train on the cleaned data. Whereas other methods decrease in performance from low sparsity (e.g., 0.0) to high sparsity (e.g. 0.6), CL methods are robust across sparsity, as indicated by comparing the two column-wise red highlighted cells.

Noise	20%				40%				70%			
Sparsity	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6
CL: $\boldsymbol{C}_{\text{confusion}}$	89.6	89.4	90.2	89.9	83.9	83.9	83.2	84.2	31.5	39.3	33.7	30.6
CL: PBC	90.5	90.1	90.6	90.7	84.8	85.5	85.3	86.2	33.7	40.7	35.1	31.4
CL: $\boldsymbol{C}_{\tilde{y},y^*}$	**91.1**	**90.9**	**91.1**	**91.3**	86.7	86.7	86.6	86.9	32.4	**41.8**	34.4	34.5
CL: C+NR	90.8	90.7	91.0	91.1	**87.1**	**86.9**	**86.7**	**87.2**	**41.1**	41.7	39.0	32.9
CL: PBNR	90.7	90.5	90.9	90.9	**87.1**	86.8	86.6	**87.2**	41.0	**41.8**	**39.1**	**36.4**
INCV (Chen et al., 2019)	87.8	88.6	89.6	89.2	84.4	76.6	85.4	73.6	28.3	25.3	34.8	29.7
Mixup (Zhang et al., 2018)	85.6	86.8	87.0	84.3	76.1	75.4	68.6	59.8	32.2	31.3	32.3	26.9
SCE-loss (Wang et al., 2019)	87.2	87.5	88.8	84.4	76.3	74.1	64.9	58.3	33.0	28.7	30.9	24.0
MentorNet (Jiang et al., 2018)	84.9	85.1	83.2	83.4	64.4	64.2	62.4	61.5	30.0	31.6	29.3	27.9
Co-Teaching (Han et al., 2018)	81.2	81.3	81.4	80.6	62.9	61.6	60.9	58.1	30.5	30.2	27.7	26.0
S-Model (Goldberger et al., 2017)	80.0	80.0	79.7	79.1	58.6	61.2	59.1	57.5	28.4	28.5	27.9	27.3
Reed (Reed et al., 2015)	78.1	78.9	80.8	79.3	60.5	60.4	61.2	58.6	29.0	29.4	29.1	26.8
Baseline	78.4	79.2	79.0	78.2	60.2	60.8	59.6	57.3	27.0	29.7	28.2	26.8

For each CL method, sparsity, and noise setting, we report the mean accuracy in Table 2, averaged over ten trials, by varying the random seed and initial weights of the neural network for training. Standard deviations are reported in Table 3 to improve readability. For each column in Table 2, the corresponding standard deviations in in Table 3 are significantly less than the performance difference between CL methods and baseline methods. Notably, all standard deviations are significantly (∼10x) less than the mean performance difference between the top-performing CL method and baseline methods for each setting, averaged over random weight initialization. Standard deviations are only reported for CL methods because of difficulty reproducing consistent results for some of the other methods.

Table 3: Standard deviations (% units) associated with the mean score (over ten trials) for scores reported for CL methods in Table 2. Each trial uses a different random seed and network weight initialization. No standard deviation exceeds 2%.

Noise	20%				40%				70%			
Sparsity	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6
CL: $\boldsymbol{C}_{\text{confusion}}$	0.07	0.10	0.17	0.08	0.19	0.22	0.23	0.20	0.93	0.24	0.13	0.26
CL: PBC	0.14	0.12	0.11	0.10	0.15	0.17	0.16	0.10	0.12	0.22	0.11	0.30
CL: $\boldsymbol{C}_{\tilde{y},y^*}$	0.17	0.09	0.17	0.11	0.10	0.20	0.09	0.13	1.02	0.15	0.18	1.63
CL: C+NR	0.09	0.10	0.08	0.08	0.11	0.14	0.16	0.10	0.42	0.33	0.26	1.90
CL: PBNR	0.15	0.09	0.09	0.10	0.18	0.10	0.15	0.12	0.26	0.28	0.24	1.43

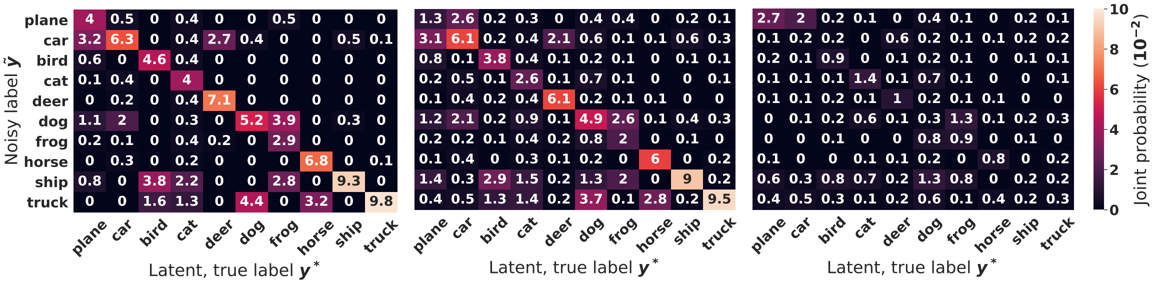
Table 4: Mean accuracy, F1, precision, and recall measures of CL methods for finding label errors in CIFAR-10, averaged over ten trials.

Measure	Accuracy (%) ± Std. Dev. (%)				F1 (%)				Precision (%)				Recall (%)			
Noise	20%		40%		20%		40%		20%		40%		20%		40%	
Sparsity	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6	0.0	0.6
CL: $\boldsymbol{C}_{\text{confusion}}$	84±0.07	85±0.09	85±0.24	81±0.21	71	72	84	79	56	58	74	70	**98**	**97**	**97**	**90**
CL: $\boldsymbol{C}_{\tilde{y},y^*}$	89±0.15	**90**±0.10	86±0.15	**84**±0.12	75	78	84	**80**	**67**	**70**	78	77	86	88	91	84
CL: PBC	88±0.22	88±0.11	86±0.17	82±0.13	76	76	84	79	64	65	76	74	96	93	94	85
CL: PBNR	89±0.11	**90**±0.08	**88**±0.12	**84**±0.11	77	**79**	**85**	**80**	65	68	**82**	**79**	93	94	88	82
CL: C+NR	**90**±0.21	**90**±0.10	87±0.23	83±0.14	**78**	78	84	78	**67**	69	**82**	**79**	93	90	87	78

We also evaluate CL's accuracy in finding label errors. In Table 4, we compare five variants of CL methods across noise and sparsity and report their precision, recall, and F1 in recovering the true label. The results show that CL is able to find the label errors with high recall and reasonable F1.

Robustness to Sparsity Table 2 reports CIFAR test accuracy for learning with noisy labels across noise amount and sparsity, where the first five rows report our CL approaches. As shown, CL consistently performs well compared to prior art across all noise and sparsity settings. We observe significant improvement in high-noise and/or high-sparsity regimes. The simplest CL method $CL : \boldsymbol{C}_{\text{confusion}}$ performs similarly to *INCV* and comparably to prior art with best performance by $\boldsymbol{C}_{\tilde{y},y^*}$ across all noise and sparsity settings. The results validate the benefit of directly modeling the joint noise distribution and show that our method is competitive compared to highly competitive, robust learning methods.

To understand why CL performs well, we evaluate CL joint estimation across noise and sparsity with RMSE in Table S1 in the Appendix and estimated $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ in Fig. S1 in the Appendix. For the 20% and 40% noise settings, on average, CL achieves an RMSE of .004 relative to the true joint $\boldsymbol{Q}_{\tilde{y},y^*}$ across all sparsities. The simplest CL variant, $\boldsymbol{C}_{\text{confusion}}$ normalized via Eqn. (3) to obtain $\hat{\boldsymbol{Q}}_{\text{confusion}}$, achieves a slightly worse RMSE of .006.



(a) True $\boldsymbol{Q}_{\tilde{y},y^*}$ (unknown to CL) (b) CL estimated $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ (c) Absolute diff. $|\boldsymbol{Q}_{\tilde{y},y^*} - \hat{\boldsymbol{Q}}_{\tilde{y},y^*}|$

Figure 2: Our estimation of the joint distribution of noisy labels and true labels for CIFAR with 40% label noise and 60% sparsity. Observe the similarity (RSME = .004) between (a) and (b) and the low absolute error in every entry in (c). Probabilities are scaled up by 100.

In Fig. 2, we visualize the quality of CL joint estimation in a challenging high-noise (40%), high-sparsity (60%) regime on CIFAR. Subfigure (a) demonstrates high sparsity in the latent true joint $\boldsymbol{Q}_{\tilde{y},y^*}$, with over half the noise in just six noise rates. Yet, as can be seen in subfigures (b) and (c), CL still estimates over 80% of the entries of $\boldsymbol{Q}_{\tilde{y},y^*}$ within an absolute difference of .005. The results empirically substantiate the theoretical bounds of Section 4.

In Table S2 (see Appendix), we report the training time required to achieve the accuracies reported in Table 2 for INCV and confident learning. As shown in Table S2, INCV training time exceeded 20 hours. In comparison, CL takes less than three hours on the same machine: an hour for cross-validation, less than a minute to find errors, and an hour to re-train.

5.2 Real-world Label Errors in ILSVRC12 ImageNet Train Dataset

Russakovsky et al. (2015) suggest label errors exist in ImageNet due to human error, but to our knowledge, few attempts have been made to find label errors in the ILSVRC 2012 training set, characterize them, or re-train without them. Here, we consider each application. We use ResNet18 and ResNet50 architectures with standard settings: 0.1 initial learning rate, 90 training epochs with 0.9 momentum.

Table 5: Ten largest non-diagonal entries in the confident joint $\boldsymbol{C}_{\tilde{y},y^*}$ for ImageNet train set used for ontological issue discovery. A duplicated class detected by CL is highlighted in red.

$\boldsymbol{C}_{\tilde{y},y^*}$	$\tilde{y}$ name	y^* name	$\tilde{y}$ nid	y^* nid	$\boldsymbol{C}_{\text{confusion}}$	$\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$
645	projectile	missile	n04008634	n03773504	494	0.00050
539	tub	bathtub	n04493381	n02808440	400	0.00042
476	breastplate	cuirass	n02895154	n03146219	398	0.00037
437	green_lizard	chameleon	n01693334	n01682714	369	0.00034
435	chameleon	green_lizard	n01682714	n01693334	362	0.00034
433	missile	projectile	n03773504	n04008634	362	0.00034
417	maillot	maillot	n03710637	n03710721	338	0.00033
416	horned_viper	sidewinder	n01753488	n01756291	336	0.00033
410	corn	ear	n12144580	n13133613	333	0.00032
407	keyboard	space_bar	n04505470	n04264628	293	0.00032

Ontological discovery for dataset curation Because ImageNet is an one-hot class dataset, the classes are required to be mutually exclusive. Using ImageNet as a case study, we observe auto-discovery of ontological issues at the class level in Table 5, operationalized by listing the 10 largest non-diagonal entries in $\boldsymbol{C}_{\tilde{y},y^*}$. For example, the class *maillot* appears twice, the existence of *is-a* relationships like *bathtub is a tub*, misnomers like *projectile* and *missile*, and unanticipated issues caused by words with multiple definitions like *corn* and *ear*. We include the baseline $\boldsymbol{C}_{\text{confusion}}$ to show that while $\boldsymbol{C}_{\text{confusion}}$ finds fewer label errors than $\boldsymbol{C}_{\tilde{y},y^*}$, they rank ontological issues similarly.

Finding label issues Fig. 3 depicts the top 16 label issues found using CL: PBNR with ResNet50 ordered by the normalized margin. We use the term *issue* versus *error* because examples found by CL consist of a mixture of multi-label images, ontological issues, and

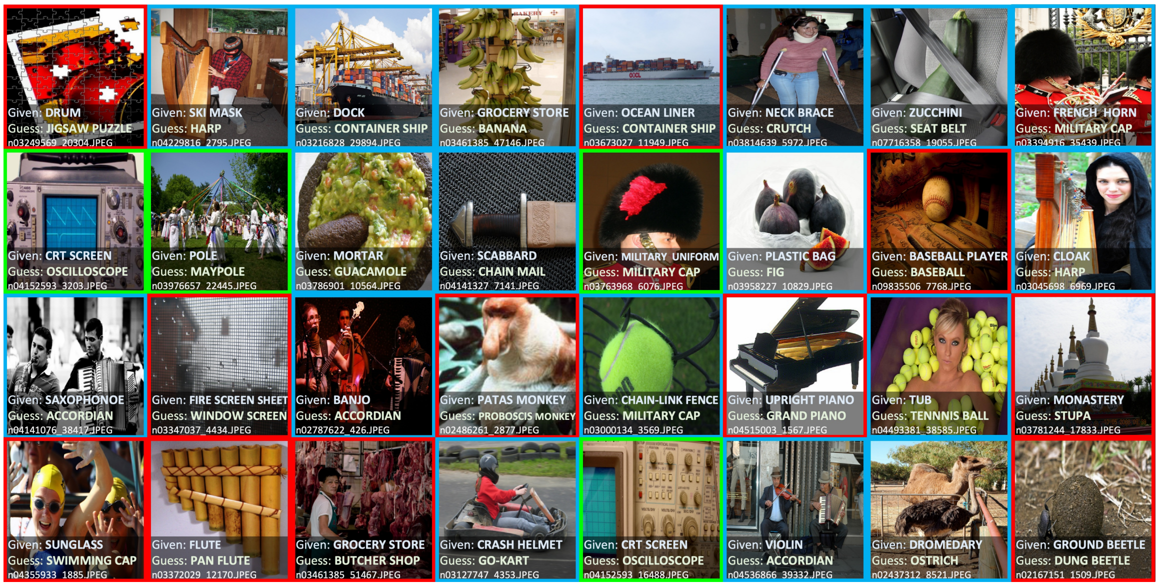
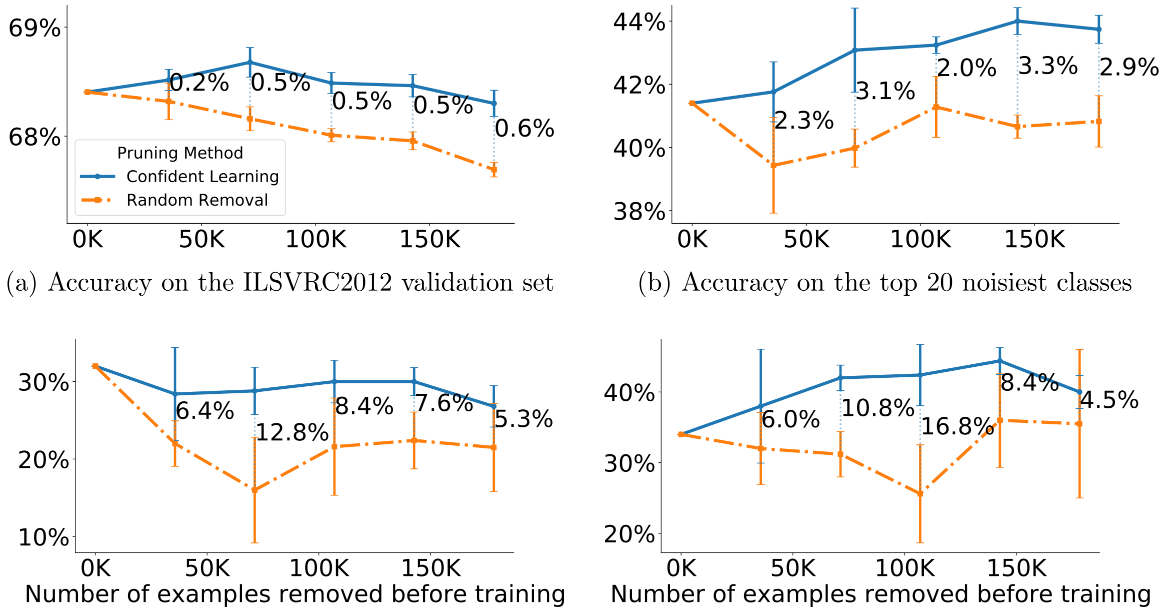


Figure 3: Top 32 (ordered automatically by normalized margin) identified label issues in the 2012 ILSVRC ImageNet train set using CL: PBNR. Errors are boxed in red. Ontological issues are boxed in green. Multi-label images are boxed in blue.

actual label errors. Examples of each are indicated by colored borders in the figure. To evaluate CL in the absence of true labels, we conducted a small-scale human validation on a random sample of 500 errors (as identified using CL: PBNR) and found 58% were either multi-label, ontological issues, or errors. ImageNet data are often presumed error-free, yet ours is the first attempt to identify label errors automatically in ImageNet training images.

Training ResNet on ImageNet with label issues removed By providing cleaned data for training, we explore how CL can be used to achieve similar or better validation accuracy on ImageNet when trained with less data. To understand the performance differences, we train ResNet-18 (Fig. 4) on progressively less data, removing 20%, 40%,..., 100% of ImageNet train set label issues identified by CL and training from scratch each time. Fig. 4 depicts the top-1 validation accuracy when training with cleaned data from CL versus removing uniformly random examples, on each of (a) the entire ILSVRC validation set, (b) the 20 (noisiest) classes with the smallest diagonal in $\boldsymbol{C}_{\tilde{y},y^*}$, (c) the foxhound class, which has the smallest diagonal in $\boldsymbol{C}_{\tilde{y},y^*}$, and (d) the maillot class, a known erroneous class, duplicated accidentally in ImageNet, as previously published (Hoffman et al., 2015), and verified (c.f. line 7 in Table 5). For readability, we plot the best performing CL method at each point and provide the individual performance of each CL method in the Appendix (see Fig. S2). For the case of a single class, as shown in Fig. 4(c) and 4(d), we show the recall using the model's top-1 prediction, hence the comparatively larger variance in classification accuracy reported compared to (a) and (b). We observed that CL outperforms the random removal baseline in nearly all experiments, and improves on the no-data-removal baseline accuracy, depicted by the left-most point in the subfigures, on average over the five trials for the 1,000 and 20 class settings, as shown in Fig. 4(a) and 4(b). To verify the result is not model-specific, we



(a) Accuracy on the ILSVRC2012 validation set

(b) Accuracy on the top 20 noisiest classes

(c) Accuracy on the noisiest class: foxhound

(d) Accuracy on known erroneous class: maillot

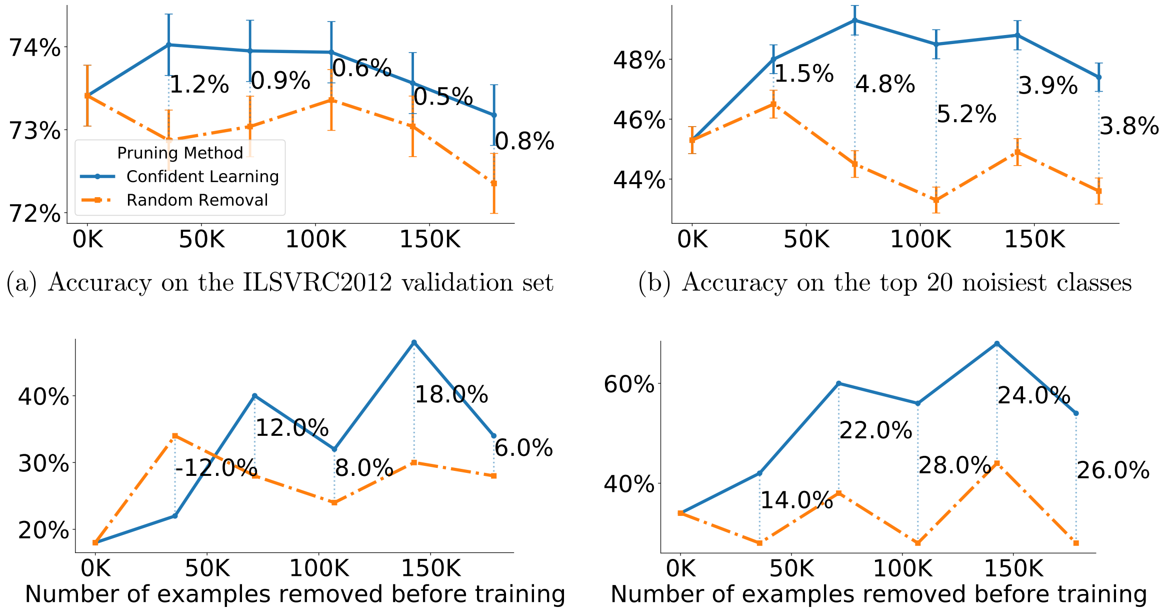
Figure 4: ResNet-18 Validation Accuracy on ImageNet (ILSVRC2012) when 20%, 40%, ..., 100% of the label issues found using confident learning are removed prior to training (blue, solid line) compared with random examples removed prior to training (orange, dash-dotted line). Each subplot is read from left-to-right as incrementally more CL-identified issues are removed prior to training (shown by the x-axis). The translucent black dotted verticals bars measure the improvement when removing examples with CL vs random examples. Each point in all subfigures represents an independent training of ResNet-18 from scratch. Each point on the graph depicts the average accuracy of 5 trials (varying random seeding and weight initialization). The capped, colored vertical bars depict the standard deviation.

repeat each experiment for a single trial with ResNet-50 (Fig. 5) and find that CL similarly outperforms the random removal baseline.

These results suggest that CL can reduce the size of a real-world noisy training dataset by 10% while still moderately improving the validation accuracy (Figures 4a, 4b, 5a, 5b) and significantly improving the validation accuracy on the erroneous maillot class (Figures 4d, 5d). While we find CL methods may improve the standard ImageNet training on clean training data by filtering out a subset of training examples, the significance of this result lies not in the magnitude of improvement, but as a warrant of exploration in the use of cleaning methods when training with ImageNet, which is typically assumed to have correct labels. Whereas many of the label issues in ImageNet are due to multi-labeled examples (Yun et al., 2021), next we consider a dataset with disjoint classes.

5.3 Amazon Reviews Dataset: CL using logistic regression on noisy text data

The Amazon Reviews dataset is a corpus of textual reviews labeled with 1-star to 5-star ratings from Amazon customers used to benchmark sentiment analysis models (He and



(a) Accuracy on the ILSVRC2012 validation set

(b) Accuracy on the top 20 noisiest classes

(c) Accuracy on the noisiest class: foxhound

(d) Accuracy on known erroneous class: maillot

Figure 5: Replication of the experiments in Fig. 4 with ResNet-50. Each point in each subfigure depicts the accuracy of a single trial (due to computational limitations). Error bars, shown by the colored vertical lines, are estimated via Clopper-Pearson intervals for subfigures (a) and (b). For additional information, see the caption of Fig. 4.

McAuley, 2016). We study the 5-core (9.9 GB) variant of the dataset – the subset of data in which all users and items have at least 5 reviews. 2-star and 4-star reviews are removed due to ambiguity with 1-star and 5-star reviews, respectively. Left in the dataset, 2-star and 4-star reviews could inflate error counts, making CL appear to be more effective than it is.

This subsection serves three goals. First, we use a logistic regression classifier, as opposed to a deep-learning model, for our experiments in this section to evaluate CL for non-deep-learning methods. Second, we seek to understand how CL may improve learning with noise in the label space of text data, but not noise in the text data itself (e.g. typos). Towards this goal, we consider non-empty reviews with more “helpful” up-votes than down-votes – the resulting dataset consists of approximately ten million reviews. Finally, Theorem 2 shows that CL is robust to class-imbalance, but datasets like ImageNet and CIFAR-10 are balanced by construction: the Amazon Reviews dataset, however, is naturally and extremely imbalanced – the distribution of given labels (i.e., the noisy prior), is: 9% 1-star reviews:, 12% 3-star reviews, and 79% 5-star reviews. We seek to understand if CL can find label errors and improve performance in learning with noisy labels in this class-imbalanced setting.

Training settings To demonstrate that non-deep-learning methods can be effective in finding label issues under the CL framework, we use a multinomial logistic regression classifier for both finding label errors and learning with noisy labels. The built-in SGD optimizer in the open-sourced fastText library (Joulin et al., 2017) is used with settings: initial learning rate = 0.1, embedding dimension = 100, and n-gram = 3). Out-of-sample predicted probabilities are obtained via 5-fold cross-validation. For input during training, a review is represented as the mean of pre-trained, tri-gram, word-level fastText embeddings (Bojanowski et al., 2017).

Finding label issues Table 6 shows examples of label issues in the Amazon Reviews dataset found automatically using the CL: C+NR variant of confident learning. We observe qualitatively that most label issues identified by CL in this context are reasonable except for sarcastic reviews, which appear to be poorly modeled by the bag-of-words approach.

Table 6: Top 20 CL-identified label issues in the Amazon Reviews text dataset using CL: C+NR, ordered by normalized margin. A logistic regression classifier trained on fastText embeddings is used to obtain out-of-sample predicted probabilities. Most errors are reasonable, with the exception of sarcastic reviews, which are poorly modeled by the bag-of-words model.

Review	**Given Label**	**CL Guess**
A very good addition to kindle. Cleans and scans. Very easy TO USE	⋆	⋆⋆⋆⋆⋆
Buy it and enjoy a great story.	⋆⋆⋆	⋆⋆⋆⋆⋆
Works great! I highly recommend it to everyone that enjoys singing hymns! Love it! Love it! Love it! :) .	⋆⋆⋆	⋆⋆⋆⋆⋆
Awesome it was better than all the other my weirder school books. I love it! The best book ever.Awesome	⋆	⋆⋆⋆⋆⋆
I gave this 5 stars under duress. I would rather give it 3 stars. it plays fine but it is a little boring so far.	⋆⋆⋆⋆⋆	⋆⋆⋆
only six words: don’t waist your money on this	⋆⋆⋆⋆⋆	⋆
I love it so much at first I though it would be boring but turns out its fun for all ages get it	⋆	⋆⋆⋆⋆⋆
Excellent read, could not put it down! Keep up the great works ms. Brown. Cannot wait to download the next one.	⋆	⋆⋆⋆⋆⋆
This is one of the easiest to use games I have ever played. It is adaptable and fun. I love it.	⋆	⋆⋆⋆⋆⋆
So this is what today’s music has become?	⋆	⋆⋆⋆⋆⋆
Sarah and Charlie, what a wonderful story. I loved this book and look forward to reading more of this series.	⋆⋆⋆	⋆⋆⋆⋆⋆
I’ve had this for over a year and it works very well. I am very happy with this purchase.	⋆	⋆⋆⋆⋆⋆
this show is insane and I love it. I will be ordering more seasons of it.	⋆⋆⋆	⋆⋆⋆⋆⋆
Just what the world needs, more generic r&b.	⋆	⋆⋆⋆⋆⋆
I did like the Making Of This Is movie it okay it not the best okay it not great .	⋆	⋆⋆⋆
Tough game. But of course it has the very best sound track ever!	⋆	⋆⋆⋆⋆⋆
unexpected kid on the way thanks to this shit	⋆	⋆⋆⋆⋆⋆
The kids are fascinated by it, Plus my wife loves it.. I love it I love it we love it	⋆⋆⋆	⋆⋆⋆⋆⋆
Loved this book! A great story and insight into the time period and life during those times. Highly recommend this book	⋆⋆⋆	⋆⋆⋆⋆⋆
Great reading I could not put it down. Highly recommend reading this book. You will not be disappointed. Must read.	⋆⋆⋆	⋆⋆⋆⋆⋆

Table 7: Ablation study (varying train set size, test split, and epochs) comparing test accuracy (%) of CL methods versus a standard training baseline for classifying noisy, real-world Amazon reviews text data as either 1-star, 3-stars, or 5-stars. A simple multinomial logistic regression classifier is used. Mean top-1 accuracy and standard deviations are reported over five trials. The number of estimated label errors CL methods removed prior to training is shown in the "Pruned" column. Baseline training begins to overfit to noise with additional epochs trained, whereas CL test accuracy continues to increase *(cf. N=1000K, Epochs: 50)*.

Test	Train set size	$N = 1000K$				$N = 500K$		
		Epochs: 5	Epochs: 20	Epochs: 50	Pruned	Epochs: 5	Epochs: 20	Pruned
10th	CL: $\boldsymbol{C}_{\text{confusion}}$	85.2±0.06	89.2±0.02	90.0±0.02	291K	86.6±0.03	86.6±0.03	259K
	CL: C+NR	86.3±0.04	**89.8±0.01**	**90.2±0.01**	250K	**87.5±0.05**	**87.5±0.03**	244K
	CL: $\boldsymbol{C}_{\tilde{y},y^*}$	**86.4±0.01**	**89.8±0.02**	90.1±0.02	246K	**87.5±0.02**	**87.5±0.02**	243K
	CL: PBC	86.2±0.03	89.7±0.01	**90.2±0.01**	257K	87.4±0.03	87.4±0.03	247K
	CL: PBNR	86.2±0.07	89.7±0.01	**90.2±0.01**	257K	87.4±0.05	87.4±0.05	247K
	Baseline	83.9±0.11	86.3±0.06	84.4±0.04	0K	82.7±0.07	82.8±0.07	0K
11th	CL: $\boldsymbol{C}_{\text{confusion}}$	85.3±0.05	89.3±0.01	90.0±0.0	294K	86.6±0.04	86.6±0.06	261K
	CL: C+NR	**86.4±0.06**	**89.8±0.01**	90.2±0.01	252K	**87.5±0.04**	**87.5±0.03**	247K
	CL: $\boldsymbol{C}_{\tilde{y},y^*}$	86.3±0.05	**89.8±0.01**	90.1±0.02	249K	**87.5±0.03**	**87.5±0.02**	246K
	CL: PBC	86.2±0.03	**89.8±0.01**	**90.3±0.0**	260K	87.4±0.03	87.4±0.05	250K
	CL: PBNR	86.2±0.06	**89.8±0.01**	90.2±0.02	260K	87.4±0.05	87.4±0.03	249K
	Baseline	83.9±0.0	86.3±0.05	84.4±0.12	0K	82.7±0.04	82.7±0.09	0K

Learning with noisy labels / weak supervision We compare the CL methods, which prune errors from the train set and subsequently provide clean data for training, versus a standard training baseline (denoted *Baseline* in Table 7), which trains on the original, uncleaned train dataset. The same training settings used to find label errors (see Subsection 5.3) are used to obtain all scores reported in Table 7 for all methods. For a fair comparison, all mean accuracies in Table 7 are reported on the same held-out test set, created by splitting the Amazon reviews dataset into a train set and test set such that every tenth example is placed in a test set and the remaining data is available for training (the Amazon Reviews 5-core dataset provides no explicit train set and test set).

The Amazon Reviews dataset is naturally noisy, but the fraction of noise in the dataset is estimated to be less than 4% (Northcutt et al., 2021), which makes studying the benefits of providing clean data for training challenging. To increase the percentage of noisy labels without adding synthetic noise, we subsample 1 million training examples from the train set by combining the label issues identified by all five CL methods from the original training data (244K examples) and a uniformly random subsample (766k examples) of the remaining "cleaner" training data. This process increases the percentage of label noise to 24% (estimated) in the train set and, importantly, does *not* increase the percentage of noisy labels in the test set – large amounts of test set label noise have been shown to severely impact benchmark rankings (Northcutt et al., 2021).

To mitigate the bias induced by the choice of train set size, test set split, and the number of epochs trained, we conduct an ablation study shown in Table 7. For the train set size, we repeat each experiment with train set sizes of 1-million examples and $500,000$ examples. For the test set split, we repeat all experiments by removing every *eleventh* example (instead of tenth) in our train/test split (c.f. the first column in Table 7), minimizing the overlap (9%) between the two test sets. For each number of epochs trained, we repeat each experiment with 5, 20, and 50 epochs. We omit ($N = 500K$, Epochs: 50) because no learning occurs after 5 epochs.

Every score reported in Table 7 is the mean and standard deviation of five trials: each trial varies the randomly selected subset of training data and the initial weights of the logistic regression model used for training.

The results in Table 7 reveal three notable observations. First, all CL methods outperform the baseline method by a significant margin in all cases. Second, CL methods outperform the baseline method even with nearly half of the training data pruned (Table 7, cf. N=500K). Finally, for the train set size $N = 1000K$, baseline training begins to overfit to noise with additional epochs trained, whereas CL test accuracy continues to increase *(cf. N=1000K, Epochs: 50)*, suggesting CL robustness to overfitting to noise during training. The results in Table 7 suggest CL's efficacy for noisy supervision with logistic regression in the context of text data.

5.4 Real-world Label Errors in Other Datasets

We use CL to find label errors in the purported "error-free" MNIST dataset comprised of preprocessed black-and-white handwritten digits, and also in the noisy-labeled WebVision dataset (Li et al., 2017a) comprised of color images collected from online image repositories and using the search query as the noisy label.

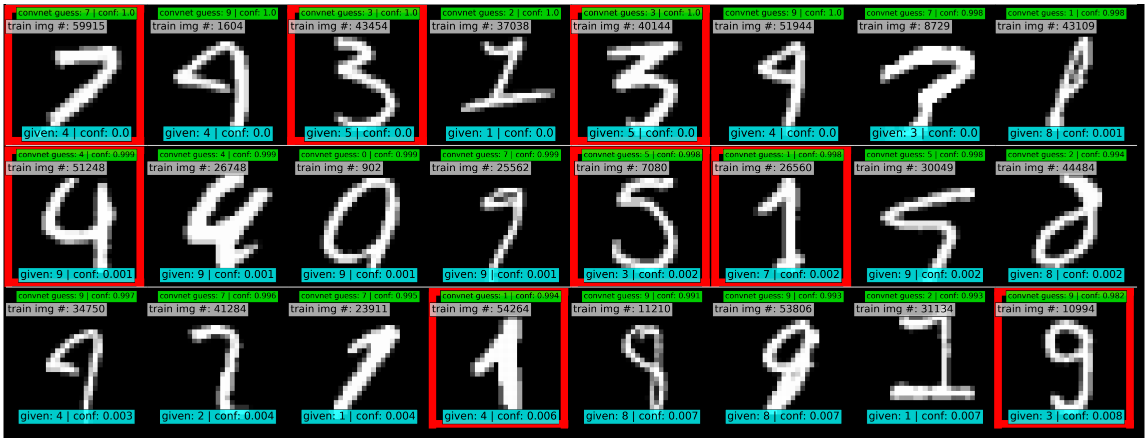


Figure 6: Label errors in the original, unperturbed MNIST train dataset identified using CL: PBNR. These are the top 24 errors found by CL, ordered left-right, top-down by increasing self-confidence, denoted *conf* in teal. The predicted $\arg\max \hat{p}(\tilde{y}=k; x, \boldsymbol{\theta})$ label is in green. Overt errors are in red. This dataset is assumed "error-free" in tens of thousands of studies.

To our surprise, the original, unperturbed MNIST dataset, which is predominately assumed error-free, contains blatant label errors, highlighted by the red boxes in Fig. 6. To find label errors in MNIST, we pre-trained a simple 2-layer CNN for 50 epochs, then used cross-validation to obtain $\hat{\boldsymbol{P}}_{k,i}$, the out-of-sample predicted probabilities for the train set. CL: PBNR was used to identify the errors. The top 24 label errors, ordered by self-confidence, are shown in Fig. 6. For verification, the indices of the train label errors are shown in grey.

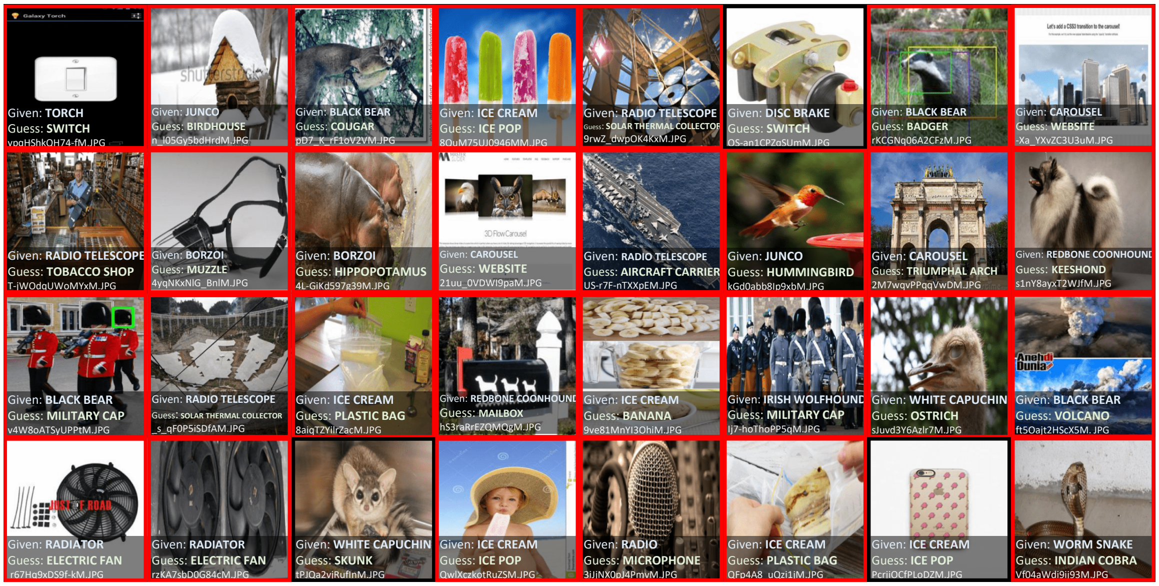


Figure 7: Top 32 identified label issues in the WebVision train set using CL: $\boldsymbol{C}_{\tilde{y},y^*}$. Out-of-sample predicted probabilities are obtained using a model pre-trained on ImageNet, avoiding training entirely. Errors are boxed in red. Ambiguous cases or mistakes are boxed in black. Label errors are ordered automatically by normalized margin.

To find label errors in WebVision, we used a pre-trained model to obtain $\hat{\boldsymbol{P}}_{k,i}$, observing two practical advantages of CL: (1) a pre-trained model can be used to obtain $\hat{\boldsymbol{P}}_{k,i}$ out-of-sample instead of cross-validation and (2) this makes CL fast. For example, finding label errors in WebVision, with over a million images and 1,000 classes, took three minutes on a laptop using a pre-trained ResNext model that had never seen the noisy WebVision train set before. We used the CL: $\boldsymbol{C}_{\tilde{y},y^*}$ method to find the label errors and ordered errors by normalized margins. Examples of WebVision label errors found by CL are shown in Fig. 7.

6. Related work

We first discuss prior work on confident learning, then review how CL relates to noise estimation and robust learning.

Confident learning Our results build on a large body of work termed "confident learning". Elkan (2001) and Forman (2005) pioneered counting approaches to estimate false positive and false negative rates for binary classification. We extend counting principles to multi-class setting. To increase robustness against epistemic error in predicted probabilities and class imbalance, Elkan and Noto (2008) introduced thresholding, but required uncorrupted positive labels. CL generalizes the use of thresholds to multi-class noisy labels. CL also reweights the loss during training to adjust priors for the data removed. This choice builds on formative works (Natarajan et al., 2013; Van Rooyen et al., 2015) which used loss reweighting to prove equivalent empirical risk minimization for learning with noisy labels. More recently, Han et al. (2019) proposed an empirical deep self-supervised learning approach to avoid probabilities by using embedding layers of a neural network. In comparison, CL is non-iterative and theoretically grounded. Lipton et al. (2018) estimate label noise using approaches based on confusion matrices and cross-validation. However, unlike CL, the former assumes a less general form of label shift than class-conditional noise. Huang et al. (2019) demonstrate the empirical efficacy of first finding label errors, then training on clean data, but the study evaluates only uniform (symmetric) and pair label noise – CL augments these empirical findings with theoretical justification for the broader class of asymmetric and class-conditional label noise.

Theory: a model-free, data-free approach Theoretical analysis with noisy labels often assumes a restricted class of models or data to disambiguate model noise from label noise. For example, Shen and Sanghavi (2019) provide theoretical guarantees for learning with noisy labels in a more general setting than CL that includes adversarial examples and noisy data, but limit their findings to generalized linear models. CL theory is model and dataset agnostic, instead restricting the magnitude of example-level noise. In a formative related approach, Xu et al. (2019) prove that using the loss function $-\log(|\det(\boldsymbol{Q}_{\tilde{y},y^*}))|$ enables noise robust training for any model and dataset, further justified by performant empirical results. Similar to confident learning, their approach hinges on the use of $\boldsymbol{Q}_{\tilde{y},y^*}$, however, they require that $\boldsymbol{Q}_{\tilde{y}|y^*}$ is invertible and estimate $\boldsymbol{Q}_{\tilde{y},y^*}$ using $\boldsymbol{C}_{\text{confusion}}$, which is sensitive to class-imbalance and heterogeneous class probability distributions (see Sec. 3.1). In Sec. 4, we show sufficient conditions in Thm. 2 where $\boldsymbol{C}_{\tilde{y},y^*}$ exactly finds label errors, regardless of each class's probability distribution.

Uncertainty quantification and label noise estimation A number of formative works developed solutions to estimate noise rates using convergence criterion (Scott, 2015), positive-unlabeled learning (Elkan and Noto, 2008), and predicted probability ratios (Northcutt et al., 2017), but are limited to binary classification. Others prove equivalent empirical risk for *binary* learning with noisy labels (Natarajan et al., 2013; Liu and Tao, 2015; Sugiyama et al., 2012) assuming noise rates are known, which is rarely true in practice. Unlike these binary approaches, CL estimates label uncertainty in the multiclass setting, where prior work often falls into five categories: (1) theoretical contributions (Katz-Samuels et al., 2019), (2) loss modification for label noise robustness (Patrini et al., 2016, 2017; Sukhbaatar et al., 2015; Van Rooyen et al., 2015), (3) deep learning and model-specific approaches (Sukhbaatar et al., 2015; Patrini et al., 2016; Jindal et al., 2016), (4) crowd-sourced labels via multiple workers (Zhang et al., 2017b; Dawid and Skene, 1979; Ratner et al., 2016), (5) factorization, distillation (Li et al., 2017b), and imputation (Amjad et al., 2018) methods, among other (Sáez et al., 2014). Unlike these approaches, CL provides a consistent estimator for exact estimation of the joint distribution of noisy and true labels directly, under practical conditions.

Label-noise robust learning Beyond the above noise estimation approaches, extensive studies have investigated training models on noisy datasets, e.g. (Beigman and Klebanov, 2009; Brodley and Friedl, 1999). Noise-robust learning is important for deep learning because modern neural networks trained on noisy labels generalize poorly on clean validation data (Zhang et al., 2017a). A notable recent rend in noise robust learning is benchmarking with symmetric label noise in which labels are uniformly flipped, e.g. (Goldberger and Ben-Reuven, 2017; Arazo et al., 2019). However, noise in real-world datasets is highly non-uniform and often sparse. For example, in ImageNet (Russakovsky et al., 2015), *missile* is likely to be mislabeled as *projectile*, but has a near-zero probability of being mislabeled as most other classes like *wool*, *ox*, or *wine*. To approximate real-world noise, an increasing number of studies examined asymmetric noise using, e.g. loss or label correction (Patrini et al., 2017; Reed et al., 2015; Goldberger and Ben-Reuven, 2017), per-example loss reweighting (Jiang et al., 2020, 2018; Shu et al., 2019), Co-Teaching (Han et al., 2018), semi-supervised learning (Hendrycks et al., 2018; Li et al., 2017b; Vahdat, 2017), symmetric cross entropy (Wang et al., 2019), and semi-supervised learning (Li et al., 2020), among others. These approaches work by introducing novel new models or insightful modifications to the loss function during training. CL takes a loss-agnostic approach, instead focusing on generating clean data for training by directly estimating of the joint distribution of noisy and true labels.

Comparison of the INCV Method and Confident Learning The INCV algorithm (Chen et al., 2019) and confident learning both estimate clean data, use cross-validation, and use aspects of confusion matrices to deal with label errors in ML workflows. Due to these similarities, we discuss four key differences between confident learning and INCV.

First, INCV errors are found using an iterative version of the $\boldsymbol{C}_{\text{confusion}}$ confident learning baseline: any example with a different given label than its argmax prediction is considered a label error. This approach, while effective (see Table 2), fails to properly count errors for class imbalance or when a model is more confident (larger or smaller probabilities on average) for certain class than others, as discussed in Section 4. To account for this class-level bias in predicted probabilities and enable robustness, confident learning uses theoretically-supported (see Section 4) thresholds (Elkan, 2001; Richard and Lippmann, 1991) while estimating the confident joint. Second, a major contribution of CL is finding the label errors in the presumed error-free benchmarks such as ImageNet and MNIST, whereas INCV emphasizes empirical results for learning with noisy labels. Third, in each INCV training iteration, 2-fold cross-validation is performed. The iterative nature of INCV makes training slow (see Appendix Table S2) and uses fewer data during training. Unlike INCV, confident learning is not iterative. In confident learning, cross-validated probabilities are computed only once beforehand from which the joint distribution of noisy and true labels is directly estimated which is used to identify clean data to be used by a single pass re-training. We demonstrate this approach is experimentally performant without iteration (see Table 2). Finally, confident learning is modular. CL approaches for training, finding label errors, and ordering label errors for removal are independent. In INCV, the procedure is iterative, and all three steps are tied together in a single looping process. A single iteration of INCV equates to the $\boldsymbol{C}_{\text{confusion}}$ baseline benchmarked in this paper.

7. Conclusion and Future Work

Following the principles of confident learning, we developed a novel approach to estimate the joint distribution of label noise and explicated theoretical and experimental insights into the benefits of doing so. We demonstrated accurate uncertainty quantification in high noise and and sparsity regimes, across multiple datasets, data modalities, and model architectures. We empirically evaluated three criteria: (1) uncertainty quantification via estimation of the joint distribution of label noise, (2) finding label errors, and (3) learning with noisy labels on CIFAR-10, and found that CL methods outperform recent prior art across all three.

These findings emphasize the practical nature of confident learning, identifying numerous pre-existing label issues in ImageNet, Amazon Reviews, MNIST, and other datasets, and improving the performance of learning models like deep neural networks by training on a cleaned dataset. Confident learning motivates the need for further understanding of dataset uncertainty estimation, methods to clean training and test sets, and approaches to identify ontological and label issues for dataset curation. Future directions include validation of CL methods on more datasets such as the OpenML Benchmark (Feurer et al., 2019), the multi-modal Egocentric Communications (EgoCom) benchmark (Northcutt et al., 2020), and the realistic noisy label benchmark CNWL (Jiang et al., 2020); evaluation of CL methods using other non-neural network models, such as random forests and XGBoost; examination of other threshold function formulations; examination of label errors in test sets and they affect machine learning benchmarks at scale (Northcutt et al., 2021); assimilation of CL label error finding with pseudo-labeling and/or curriculum learning to *dynamically* provide clean data during training; and further exploration of iterative and/or regression-based extensions of CL methods.

Acknowledgements

We thank the following colleagues: Jonas Mueller assisted with notation. Anish Athayle suggested starting the proof in claim 1 of Theorem 1 with the identity. Tailin Wu contributed to Lemma 1. Niranjan Subrahmanya provided feedback on baselines for confident learning.

References

Amjad, M., Shah, D., and Shen, D. (2018). Robust synthetic control. *Journal of Machine Learning Research (JMLR)*, 19(1):802–852.

Angluin, D. and Laird, P. (1988). Learning from noisy examples. *Machine Learning*, 2(4):343–370.

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. (2019). Unsupervised label noise modeling and loss correction. In *International Conference on Machine Learning (ICML)*.

Beigman, E. and Klebanov, B. B. (2009). Learning with annotation noise. In *Annual Conference of the Association for Computational Linguistics (ACL)*.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Bouguelia, M.-R., Nowaczyk, S., Santosh, K., and Verikas, A. (2018). Agreeing to disagree: active learning with noisy labels without crowdsourcing. *International Journal of Machine Learning and Cybernetics*, 9(8):1307–1319.

Brodley, C. E. and Friedl, M. A. (1999). Identifying mislabeled training data. *Journal of Artificial Intelligence Research (JAIR)*, 11:131–167.

Chen, P., Liao, B. B., Chen, G., and Zhang, S. (2019). Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning (ICML)*.

Chowdhary, K. and Dupuis, P. (2013). Distinguishing and integrating aleatoric and epistemic variation in uncertainty quantification. *Mathematical Modelling and Numerical Analysis (ESAIM)*, 47(3):635–662.

Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28.

Elkan, C. (2001). The foundations of cost-sensitive learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Elkan, C. and Noto, K. (2008). Learning classifiers from only positive and unlabeled data. In *SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*.

Feurer, M., van Rijn, J. N., Kadra, A., Gijsbers, P., Mallik, N., Ravi, S., Müller, A., Vanschoren, J., and Hutter, F. (2019). Openml-python: an extensible python api for openml. *arXiv preprint arXiv:1911.02490*.

Forman, G. (2005). Counting positives accurately despite inaccurate classification. In *European Conference on Computer Vision (ECCV)*.

Forman, G. (2008). Quantifying counts and costs via classification. *Data Mining and Knowledge Discovery*, 17(2):164–206.

Goldberger, J. and Ben-Reuven, E. (2017). Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations (ICLR)*.

Graepel, T. and Herbrich, R. (2001). The kernel gibbs sampler. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*.

Halpern, Y., Horng, S., Choi, Y., and Sontag, D. (2016). Electronic medical record phenotyping using the anchor and learn framework. *Journal of the American Medical Informatics Association*, 23(4):731–740.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. (2018). Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Han, J., Luo, P., and Wang, X. (2019). Deep self-learning from noisy labels. In *International Conference on Computer Vision (ICCV)*.

He, R. and McAuley, J. (2016). Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *International conference on world wide web (WWW)*.

Hendrycks, D. and Gimpel, K. (2017). A baseline for detecting misclassified and out-of-distribution examples in neural networks. *International Conference on Learning Representations (ICLR)*.

Hendrycks, D., Mazeika, M., Wilson, D., and Gimpel, K. (2018). Using trusted data to train deep networks on labels corrupted by severe noise. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Hoffman, J., Pathak, D., Darrell, T., and Saenko, K. (2015). Detector discovery in the wild: Joint multiple instance and representation learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, J., Qu, L., Jia, R., and Zhao, B. (2019). O2u-net: A simple noisy label detection approach for deep neural networks. In *International Conference on Computer Vision (ICCV)*.

Jiang, L., Huang, D., Liu, M., and Yang, W. (2020). Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning (ICML)*.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. (2018). Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning (ICML)*.

Jindal, I., Nokleby, M., and Chen, X. (2016). Learning deep networks from noisy labels with dropout regularization. In *International Conference on Data Mining (ICDM)*.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Annual Conference of the Association for Computational Linguistics (ACL)*.

Katz-Samuels, J., Blanchard, G., and Scott, C. (2019). Decontamination of mutual contamination models. *Journal of Machine Learning Research (JMLR)*, 20(41):1–57.

Khetan, A., Lipton, Z. C., and Anandkumar, A. (2018). Learning from noisy singly-labeled data. In *International Conference on Learning Representations (ICLR)*.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*.

Lawrence, N. D. and Schölkopf, B. (2001). Estimating a kernel fisher discriminant in the presence of label noise. In *International Conference on Machine Learning (ICML)*.

Li, J., Socher, R., and Hoi, S. C. (2020). Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations (ICLR)*.

Li, W., Wang, L., Li, W., Agustsson, E., and Van Gool, L. (2017a). Webvision database: Visual learning and understanding from web data. *arXiv:1708.02862*.

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, L.-J. (2017b). Learning from noisy labels with distillation. In *International Conference on Computer Vision (ICCV)*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*.

Lipton, Z., Wang, Y.-X., and Smola, A. (2018). Detecting and correcting for label shift with black box predictors. In *International Conference on Machine Learning (ICML)*.

Liu, T. and Tao, D. (2015). Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(3):447–461.

Natarajan, N., Dhillon, I. S., Ravikumar, P., and Tewari, A. (2017). Cost-sensitive learning with noisy labels. *Journal of Machine Learning Research (JMLR)*, 18:155–1.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. (2013). Learning with noisy labels. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Northcutt, C., Zha, S., Lovegrove, S., and Newcombe, R. (2020). Egocom: A multi-person multi-modal egocentric communications dataset. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*.

Northcutt, C. G., Athalye, A., and Mueller, J. (2021). Pervasive label errors in test sets destabilize machine learning benchmarks. In *International Conference on Learning Representations Workshop Track (ICLR)*.

Northcutt, C. G., Ho, A. D., and Chuang, I. L. (2016). Detecting and preventing “multiple-account” cheating in massive open online courses. *Computers & Education*, 100:71–80.

Northcutt, C. G., Wu, T., and Chuang, I. L. (2017). Learning with confident examples: Rank pruning for robust classification with noisy labels. In *Conference on Uncertainty in Artificial Intelligence (UAI)*.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1997). Pagerank: Bringing order to the web. Technical report, Stanford Digital Libraries Working Paper.

Patrini, G., Nielsen, F., Nock, R., and Carioni, M. (2016). Loss factorization, weakly supervised learning and label noise robustness. In *International Conference on Machine Learning (ICML)*.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and Qu, L. (2017). Making deep neural networks robust to label noise: A loss correction approach. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ratner, A. J., De Sa, C. M., Wu, S., Selsam, D., and Ré, C. (2016). Data programming: Creating large training sets, quickly. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Reed, S. E., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. (2015). Training deep neural networks on noisy labels with bootstrapping. In *International Conference on Learning Representations (ICLR)*.

Richard, M. D. and Lippmann, R. P. (1991). Neural network classifiers estimate bayesian a posteriori probabilities. *Neural computation*, 3(4):461–483.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

Sáez, J. A., Galar, M., Luengo, J., and Herrera, F. (2014). Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition. *Knowledge and Information Systems*, 38(1):179–206.

Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., and Aroyo, L. M. (2021). "Everyone wants to do the model work, not the data work": Data cascades in high-stakes ai. In *Conference on Human Factors in Computing Systems (CHI)*.

Scott, C. (2015). A rate of convergence for mixture proportion estimation, with application to learning from noisy labels. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Shen, Y. and Sanghavi, S. (2019). Learning with bad training data via iterative trimmed loss minimization. In *International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*.

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. (2019). Meta-weight-net: Learning an explicit mapping for sample weighting. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Sugiyama, M., Suzuki, T., and Kanamori, T. (2012). *Density Ratio Estimation in ML*. Cambridge University Press, New York, NY, USA, 1st edition.

Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., and Fergus, R. (2015). Training convolutional networks with noisy labels. In *International Conference on Learning Representations (ICLR)*.

Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D. C., and Silberman, N. (2019a). Learning from noisy labels by regularized estimation of annotator confusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tanno, R., Saeedi, A., Sankaranarayanan, S., Alexander, D. C., and Silberman, N. (2019b). Learning from noisy labels by regularized estimation of annotator confusion. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vahdat, A. (2017). Toward robustness against label noise in training deep discriminative neural networks. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Van Rooyen, B., Menon, A., and Williamson, R. C. (2015). Learning with symmetric label noise: The importance of being unhinged. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. (2019). Symmetric cross entropy for robust learning with noisy labels. In *International Conference on Computer Vision (ICCV)*.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. (2018). On the margin theory of feedforward neural networks. *Computing Research Repository (CoRR)*.

Xu, Y., Cao, P., Kong, Y., and Wang, Y. (2019). L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Yun, S., Oh, S. J., Heo, B., Han, D., Choe, J., and Chun, S. (2021). Re-labeling imagenet: from single to multi-labels, from global to localized labels. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017a). Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2018). mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*.

Zhang, J., Sheng, V. S., Li, T., and Wu, X. (2017b). Improving crowdsourced label quality using noise correction. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1675–1688.

Appendix A. Theorems and proofs for confident learning

In this section, we restate the main theorems for confident learning and provide their proofs.

Lemma 1 (Ideal Thresholds)**.** *For a noisy dataset* $\boldsymbol{X} := (\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ *and model* $\boldsymbol{\theta}$*, if* $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ *is* ideal*, then* $\forall i{\in}[m], t_i = \sum_{j\in[m]} p(\tilde{y} = i | y^*{=}j) p(y^*{=}j | \tilde{y} = i)$.

Proof. We use t_i to denote the thresholds used to partition $\boldsymbol{X}$ into m bins, each estimating one of $\boldsymbol{X}_{y^*}$. By definition,

$$\forall i{\in}[m], t_i = \mathbb{E}_{\boldsymbol{x}\in\boldsymbol{X}_{\tilde{y}=i}}\, \hat{p}(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta})$$

For any t_i, we show the following.

$$t_i = \mathop{\mathbb{E}}_{\boldsymbol{x}\in\boldsymbol{X}_{\tilde{y}=i}} \sum_{j\in[m]} \hat{p}(\tilde{y}{=}i|y^*{=}j; \boldsymbol{x}, \boldsymbol{\theta}) \hat{p}(y^*{=}j; \boldsymbol{x}, \boldsymbol{\theta}) \quad \triangleright \text{Bayes Rule}$$

$$t_i = \mathop{\mathbb{E}}_{\boldsymbol{x}\in\boldsymbol{X}_{\tilde{y}=i}} \sum_{j\in[m]} \hat{p}(\tilde{y}{=}i|y^*{=}j) \hat{p}(y^*{=}j; \boldsymbol{x}, \boldsymbol{\theta}) \quad \triangleright \text{Class-conditional Noise Process (CNP)}$$

$$t_i = \sum_{j\in[m]} \hat{p}(\tilde{y}{=}i|y^*{=}j) \mathop{\mathbb{E}}_{\boldsymbol{x}\in\boldsymbol{X}_{\tilde{y}=i}} \hat{p}(y^*{=}j; \boldsymbol{x}, \boldsymbol{\theta})$$

$$t_i = \sum_{j\in[m]} p(\tilde{y} = i | y^* = j) p(y^* = j | \tilde{y} = i) \quad \triangleright \text{Ideal Condition}$$

This form of the threshold is intuitively reasonable: the contributions to the sum when $i = j$ represents the probabilities of correct labeling, whereas when $i \neq j$, the terms give the probabilities of mislabeling $p(\tilde{y} = i | y^* = j)$, weighted by the probability $p(y^* = j | \tilde{y} = i)$ that the mislabeling is corrected. □

Theorem 1 (Exact Label Errors)**.** *For a noisy dataset,* $\boldsymbol{X} := (\boldsymbol{x}, \tilde{y})^n {\in} (\mathbb{R}^d, [m])^n$ *and model* $\boldsymbol{\theta}{:}\boldsymbol{x}{\to}\hat{p}(\tilde{y})$*, if* $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ *is* ideal *and each diagonal entry of* $\boldsymbol{Q}_{\tilde{y}|y^*}$ *maximizes its row and column, then* $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ *and* $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$ *(consistent estimator for* $\boldsymbol{Q}_{\tilde{y},y^*}$*).*

Proof. Alg. 1 defines the construction of the confident joint. We consider Case 1: when there are collisions (trivial by the construction of Alg. 1) and case 2: when there are no collisions (harder).

Case 1 (collisions):
When a collision occurs, by the construction of the confident joint (Eqn. 1), a given example $\boldsymbol{x}_k$ gets assigned bijectively into bin

$$\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y},y^*}[\tilde{y}_k][\mathop{\arg\max}_{i\in[m]} \hat{p}(\tilde{y} = i; \boldsymbol{x}, \boldsymbol{\theta})]$$

Because we have that $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ is ideal, we can rewrite this as

$$\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y},y^*}[\tilde{y}_k][\mathop{\arg\max}_{i\in[m]} \hat{p}(\tilde{y} = i | y^*{=}y_k^*; \boldsymbol{x})]$$

And because by assumption each diagonal entry in $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its column, we have

$$\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y},y^*}[\tilde{y}_k][y^*_k]$$

Thus, any example $\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ having a collision will be exactly assigned to $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$.

Case 2 (no collisions):

We want to show that $\forall i{\in}[m], j{\in}[m], \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$.
We can partition $\boldsymbol{X}_{\tilde{y}=i}$ as

$$\boldsymbol{X}_{\tilde{y}=i} = \boldsymbol{X}_{\tilde{y}=i,y^*=j} \cup \boldsymbol{X}_{\tilde{y}=i,y^*\neq j}$$

We prove $\forall i{\in}[m], j{\in}[m], \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ by proving two claims:
Claim 1: $\boldsymbol{X}_{\tilde{y}=i,y^*=j} \subseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$
Claim 2: $\boldsymbol{X}_{\tilde{y}=i,y^*\neq j} \nsubseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$

We do not need to show $\boldsymbol{X}_{\tilde{y}\neq i,y^*=j} \nsubseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ and $\boldsymbol{X}_{\tilde{y}\neq i,y^*\neq j} \nsubseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ because the noisy labels $\tilde{y}$ are given, thus the confident joint (Eqn. 1) will never place them in the wrong bin of $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$. Thus, claim 1 and claim 2 suffice to show that $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$.

***Proof (Claim 1) of Case 2*:** Inspecting Eqn. (1) and Alg (1), by the construction of $\boldsymbol{C}_{\tilde{y},y^*}$, we have that $\forall \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i},\ \hat{p}(\tilde{y}=j|y^*{=}j;\boldsymbol{x},\boldsymbol{\theta}) \geq t_j \longrightarrow \boldsymbol{X}_{\tilde{y}=i,y^*=j} \subseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$. When the left-hand side is true, all examples with noisy label i and hidden, true label j are counted in $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$.

Thus, it suffices to prove:

$$\forall \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i}, \hat{p}(\tilde{y}=j|y^*{=}j;\boldsymbol{x},\boldsymbol{\theta}) \geq t_j \tag{5}$$

Because the predicted probabilities satisfy the ideal condition, $\hat{p}(\tilde{y}=j|y^*{=}j,\boldsymbol{x}) = p(\tilde{y}=j|y^*{=}j), \forall \boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i}$. Note the change from predicted probability, $\hat{p}$, to an exact probability, p. Thus by the ideal condition, the inequality in (5) can be written as $p(\tilde{y}=j|y^*{=}j) \geq t_j$, which we prove below:

$$\begin{aligned}
p(\tilde{y}=j|y^*{=}j) &\geq p(\tilde{y}=j|y^*{=}j)\cdot 1 && \triangleright \text{Identity}\\
&\geq p(\tilde{y}=j|y^*{=}j)\cdot \sum_{i\in[m]} p(y^*{=}i|\tilde{y}{=}j) && \\
&\geq \sum_{i\in[m]} p(\tilde{y}=j|y^*{=}j)\cdot p(y^*{=}i|\tilde{y}{=}j) && \triangleright \text{move product into sum}\\
&\geq \sum_{i\in[m]} p(\tilde{y}=j|y^*{=}i)\cdot p(y^*{=}i|\tilde{y}{=}j) && \triangleright \text{diagonal entry maximizes row}\\
&\geq t_j && \triangleright \text{Lemma 1, ideal condition}
\end{aligned}$$

***Proof (Claim 2) of Case 2*:** We prove $\boldsymbol{X}_{\tilde{y}=i,y^*\neq j} \nsubseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ by contradiction. Assume there exists some example $\boldsymbol{x}_k \in \boldsymbol{X}_{\tilde{y}=i,y^*=z}$ for $z \neq j$ such that $\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$. By claim 1, we have that $\boldsymbol{X}_{\tilde{y}=i,y^*=j} \subseteq \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$, therefore, $\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=z}$.

Thus, for some example $\boldsymbol{x}_k$, we have that $\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}$ and also $\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=z}$.

However, this is a collision and when a collision occurs, the confident joint will break the tie with arg max. Because each diagonal entry of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its row and column this will always be assign $\boldsymbol{x}_k \in \hat{\boldsymbol{X}}_{\tilde{y},y^*}[\tilde{y}_k][y^*_k]$ (the assignment from Claim 1).

This theorem also states $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$. This directly follows directly from the fact that $\forall i{\in}[m], j{\in}[m], \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$, i.e. the confident joint *exactly counts* the partitions $\boldsymbol{X}_{\tilde{y}=i,y^*=j}$ for all pairs $(i,j) \in [m] \times M$, thus $\boldsymbol{C}_{\tilde{y},y^*} = n\boldsymbol{Q}_{\tilde{y},y^*}$ and $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$. Omitting discretization error, the confident joint $\boldsymbol{C}_{\tilde{y},y^*}$, when normalized to $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$, is an exact estimator for $\boldsymbol{Q}_{\tilde{y},y^*}$. For example, if the noise rate is 0.39, but the dataset has only 5 examples in that class, the best possible estimate by removing errors is $2/5 = 0.4 \approxeq 0.39$.

□

Corollary 1.0 (Exact Estimation). *For a noisy dataset,* $(\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ *and* $\boldsymbol{\theta}{:}\boldsymbol{x}{\to}\hat{p}(\tilde{y})$, *if* $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ *is ideal and each diagonal entry of* $\boldsymbol{Q}_{\tilde{y}|y^*}$ *maximizes its row and column, and if* $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$, *then* $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$.

Proof. The result follows directly from Theorem 1. Because the confident joint *exactly counts* the partitions $\boldsymbol{X}_{\tilde{y}=i,y^*=j}$ for all pairs $(i,j) \in [m] \times M$ by Theorem 1, $\boldsymbol{C}_{\tilde{y},y^*} = n\boldsymbol{Q}_{\tilde{y},y^*}$, omitting discretization rounding errors. □

In the main text, Theorem 1 includes Corollary 1.0 for brevity. We have separated out Corollary 1.0 here to make apparent that the primary contribution of Theorem 1 is to prove $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$, from which the result of Corollary 1.0, namely that $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$ naturally follows, omitting discretization rounding errors.

Corollary 1.1 (Per-Class Robustness). *For a noisy dataset,* $\boldsymbol{X} \coloneqq (\boldsymbol{x}, \tilde{y})^n{\in}(\mathbb{R}^d, [m])^n$ *and model* $\boldsymbol{\theta}{:}\boldsymbol{x}{\to}\hat{p}(\tilde{y})$, *if* $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ *is* ***per-class diffracted*** *without label collisions and each diagonal entry of* $\boldsymbol{Q}_{\tilde{y}|y^*}$ *maximizes its row, then* $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ *and* $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$.

Proof. Re-stating the meaning of **per-class diffracted**, we wish to show that if $\hat{p}(\tilde{y}; \boldsymbol{x}, \boldsymbol{\theta})$ is diffracted with class-conditional noise s.t. $\forall j{\in}[m], \hat{p}(\tilde{y} = j; \boldsymbol{x}, \boldsymbol{\theta}) = \epsilon_j^{(1)} \cdot p^*(\tilde{y} = j | y^*{=}y^*_k) + \epsilon_j^{(2)}$ where $\epsilon_j^{(1)} \in \mathcal{R}, \epsilon_j^{(2)} \in \mathcal{R}$ (for any distribution) without label collisions and each diagonal entry of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its row, then $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ and $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$.

First note that combining linear combinations of real-valued $\epsilon_j^{(1)}$ and $\epsilon_j^{(2)}$ with the probabilities of class j for each example may result in some examples having $\hat{p}_{\boldsymbol{x},\tilde{y}=j} = \epsilon_j^{(1)} p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)} > 1$ or $\hat{p}_{\boldsymbol{x},\tilde{y}=j} = \epsilon_j^{(1)} p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)} < 0$. The proof makes no assumption about the validity of the model outputs and therefore holds when this occurs. Furthermore, confident learning does not require valid probabilities when finding label errors because confident learning depends on the *rank* principle, i.e., the rankings of the probabilities, not the values of the probabilities.

When there are no label collisions, the bins created by the confident joint are:

$$\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \coloneqq \{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \hat{p}(\tilde{y} = j; \boldsymbol{x}, \boldsymbol{\theta}) \geq t_j\} \tag{6}$$

where

$$t_j = \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \hat{p}_{\boldsymbol{x},\tilde{y}=j}$$

WLOG: we re-formulate the error $\epsilon_j^{(1)} p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)}$ as $\epsilon_j^{(1)} (p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)})$.

Now, for diffracted (non-ideal) probabilities, we rearrange how the threshold t_j changes for a given $\epsilon_j^{(1)}, \epsilon_j^{(2)}$:

$$\begin{aligned}
t_j^{\epsilon_j} &= \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \epsilon_j^{(1)} (p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)}) \\
t_j^{\epsilon_j} &= \epsilon_j^{(1)} \left(\mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} p^*_{\boldsymbol{x},\tilde{y}=j} + \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \epsilon_j^{(2)} \right) \\
t_j^{\epsilon_j} &= \epsilon_j^{(1)} \left(t_j^* + \epsilon_j^{(2)} \cdot \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} 1 \right) \\
t_j^{\epsilon_j} &= \epsilon_j^{(1)} (t_j^* + \epsilon_j^{(2)})
\end{aligned}$$

Thus, for per-class diffracted (non-ideal) probabilities, Eqn. (6) becomes

$$\begin{aligned}
\hat{\boldsymbol{X}}^{\epsilon_j}_{\tilde{y}=i,y^*=j} &= \{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \epsilon_j^{(1)} (p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_j^{(2)}) \geq \epsilon_j^{(1)} (t_j^* + \epsilon_j^{(2)})\} \\
&= \{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j^*\} \\
&= \boldsymbol{X}_{\tilde{y}=i,y^*=j} & \triangleright \text{by Theorem (1)}
\end{aligned}$$

In the second to last step, we see that the formulation of the label errors is the formulation of $\boldsymbol{C}_{\tilde{y},y^*}$ for *ideal* probabilities, which we proved yields exact label errors and consistent estimation of $\boldsymbol{Q}_{\tilde{y},y^*}$ in Theorem 1, which concludes the proof. Note that we eliminate the need for the assumption that each diagonal entry of $\boldsymbol{Q}_{\tilde{y}|y^*}$ maximizes its column because this assumption is only used in the proof of Theorem 1 when collisions occur, but here we only consider the case when there are no collisions.

□

Theorem 2 (Per-Example Robustness)**.** *For a noisy dataset,* $\boldsymbol{X} \coloneqq (\boldsymbol{x}, \tilde{y})^n \in (\mathbb{R}^d, [m])^n$ *and model* $\boldsymbol{\theta}{:}\boldsymbol{x}{\to}\hat{p}(\tilde{y})$*, if* $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ *is* ***per-example diffracted*** *without label collisions and each diagonal entry of* $\boldsymbol{Q}_{\tilde{y}|y^*}$ *maximizes its row, then* $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \approxeq \boldsymbol{X}_{\tilde{y}=i,y^*=j}$ *and* $\hat{\boldsymbol{Q}}_{\tilde{y},y^*} \approxeq \boldsymbol{Q}_{\tilde{y},y^*}$*.*

Proof. We consider the nontrivial real-world setting when a learning model $\boldsymbol{\theta}{:}\boldsymbol{x}{\to}\hat{p}(\tilde{y})$ outputs erroneous, non-ideal predicted probabilities with an error term added for every example, across every class, such that $\forall \boldsymbol{x} \in \boldsymbol{X}, \forall j \in [m]$, $\hat{p}_{\boldsymbol{x},\tilde{y}=j} = p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j}$. As a notation reminder $p^*_{\boldsymbol{x},\tilde{y}=j}$ is shorthand for the ideal probabilities $p^*(\tilde{y}=j|y^*=y_k^*) + \epsilon_{\boldsymbol{x},\tilde{y}=j}$ and $\hat{p}_{\boldsymbol{x},\tilde{y}=j}$ is shorthand for the predicted probabilities $\hat{p}(\tilde{y}=j; \boldsymbol{x}, \boldsymbol{\theta})$.

The predicted probability error $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ is distributed uniformly with no other constraints. We use $\epsilon_j \in \mathcal{R}$ to represent the mean of $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ per class, i.e. $\epsilon_j = \mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}} \epsilon_{\boldsymbol{x},\tilde{y}=j}$, which can be seen by looking at the form of the uniform distribution in Eqn. (4). If we wanted, we could add the constraint that $\epsilon_j = 0, \forall j \in [m]$ which would simplify the theorem and the proof, but is not as general and we prove exact label error and joint estimation without this constraint.

We re-iterate the form of the error in Eqn. (4) here ($\mathcal{U}$ denotes a uniform distribution):

$$\epsilon_{\boldsymbol{x},\tilde{y}=j} \sim \begin{cases} U(\epsilon_j + t_j - p^*_{\boldsymbol{x},\tilde{y}=j} \,,\, \epsilon_j - t_j + p^*_{\boldsymbol{x},\tilde{y}=j}] & p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j \\ \mathcal{U}[\epsilon_j - t_j + p^*_{\boldsymbol{x},\tilde{y}=j} \,,\, \epsilon_j + t_j - p^*_{\boldsymbol{x},\tilde{y}=j}) & p^*_{\boldsymbol{x},\tilde{y}=j} < t_j \end{cases}$$

When there are no label collisions, the bins created by the confident joint are:

$$\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \coloneqq \{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=i} : \hat{p}_{\boldsymbol{x},\tilde{y}=j} \geq t_j\} \tag{7}$$

where

$$t_j = \frac{1}{|X_{\tilde{y}=j}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \hat{p}_{\boldsymbol{x},\tilde{y}=j}$$

Rewriting the threshold t_j to include the error terms $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ and ϵ_j, we have

$$\begin{aligned} t_j^{\epsilon_j} &= \frac{1}{|X_{\tilde{y}=j}|} \sum_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \\ t_j^{\epsilon_j} &= \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} p^*_{\boldsymbol{x},\tilde{y}=j} + \mathop{\mathbb{E}}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \epsilon_{\boldsymbol{x},\tilde{y}=j} \\ &= t_j + \epsilon_j \end{aligned}$$

where the last step uses the fact that $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ is uniformly distributed over $\boldsymbol{x} \in \boldsymbol{X}$ and $n \to \infty$ so that $\mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}} \epsilon_{\boldsymbol{x},\tilde{y}=j} = \mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}} \epsilon_{\boldsymbol{x},\tilde{y}=j} = \epsilon_j$. We now complete the proof by showing that

$$p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq t_j + \epsilon_j \iff p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j$$

If this statement is true then the subsets created by the confident joint in Eqn. 7 are unaltered and therefore $\hat{\boldsymbol{X}}^{\epsilon_{\boldsymbol{x},\tilde{y}=j}}_{\tilde{y}=i,y^*=j} = \hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} \overset{Thm.\ 1}{=} \boldsymbol{X}_{\tilde{y}=i,y^*=j}$, where $\hat{\boldsymbol{X}}^{\epsilon_{\boldsymbol{x},\tilde{y}=j}}_{\tilde{y}=i,y^*=j}$ denotes the confident joint subsets for $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ predicted probabilities.

Now we complete the proof. From the distribution for $\epsilon_{\boldsymbol{x},\tilde{y}=j}$ (Eqn. 4) , we have that

$$\begin{aligned} p^*_{\boldsymbol{x},\tilde{y}=j} < t_j &\implies \epsilon_{\boldsymbol{x},\tilde{y}=j} < \epsilon_j + t_j - p^*_{\boldsymbol{x},\tilde{y}=j} \\ p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j &\implies \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq \epsilon_j + t_j - p^*_{\boldsymbol{x},\tilde{y}=j} \end{aligned}$$

Re-arranging

$$\begin{aligned} p^*_{\boldsymbol{x},\tilde{y}=j} < t_j &\implies p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} < t_j + \epsilon_j \\ p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j &\implies p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq t_j + \epsilon_j \end{aligned}$$

Using the contrapositive, we have

$$\begin{aligned} p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq t_j + \epsilon_j &\implies p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j \\ p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j &\implies p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq t_j + \epsilon_j \end{aligned}$$

Combining, we have

$$p^*_{\boldsymbol{x},\tilde{y}=j} + \epsilon_{\boldsymbol{x},\tilde{y}=j} \geq t_j + \epsilon_j \iff p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j$$

Therefore,

$$\hat{\boldsymbol{X}}^{\epsilon_{\boldsymbol{x},\tilde{y}=j}}_{\tilde{y}=i,y^*=j} \overset{Thm.\ 1}{=} \boldsymbol{X}_{\tilde{y}=i,y^*=j}$$

The last line follows from the fact that we have reduced $\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}^{\epsilon_{\boldsymbol{x}},\tilde{y}=j}$ to counting the same condition ($p^*_{\boldsymbol{x},\tilde{y}=j} \geq t_j$) as the confident joint counts under ideal probabilities in Thm (1). Thus, we maintain exact finding of label errors and exact estimation (Corollary 1.1) holds under no label collisions. The proof applies for finite datasets because we ignore discretization error, however, for equality, the proof requires the assumption $n \rightarrow \infty$ which is used in this step: $\mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}_{\tilde{y}=j}}\, \epsilon_{\boldsymbol{x},\tilde{y}=j} \overset{n \rightarrow \infty}{=} \mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}}\, \epsilon_{\boldsymbol{x},\tilde{y}=j} = \epsilon_j$. Thus, we use approximately equals in the statement of the theorem.

Note that while we use a uniform distribution in Eqn. 4, any bounded symmetric distribution with mode $\epsilon_j = \mathbb{E}_{\boldsymbol{x} \in \boldsymbol{X}}\, \epsilon_{\boldsymbol{x},j}$ is sufficient. Observe that the bounds of the distribution are non-vacuous (they do not collapse to a single value e_j) because $t_j \neq p^*_{\boldsymbol{x},\tilde{y}=j}$ by Lemma 1.

□

Algorithm 1 (**Confident Joint**) for class-conditional label noise characterization.

```
input $\hat{\boldsymbol{P}}$ an $n \times m$ matrix of out-of-sample predicted probabilities $\hat{\boldsymbol{P}}[i][j] := \hat{p}(\tilde{y}=j; x, \boldsymbol{\theta})$
input $\tilde{\boldsymbol{y}} \in \mathbb{N}_{\geq 0}{}^n$, an $n \times 1$ array of noisy labels
procedure ConfidentJoint($\hat{\boldsymbol{P}}$, $\tilde{\boldsymbol{y}}$):
PART 1 (Compute thresholds)
for $j \leftarrow 1, m$ do
    for $i \leftarrow 1, n$ do
        $l \leftarrow$ new empty list []
        if $\tilde{\boldsymbol{y}}[i] = j$ then
            append $\hat{\boldsymbol{P}}[i][j]$ to $l$
    $\boldsymbol{t}[j] \leftarrow$ average($l$)        ▷ May use percentile instead of average for more confidence
PART 2 (Compute confident joint)
$\boldsymbol{C} \leftarrow m \times m$ matrix of zeros
for $i \leftarrow 1, n$ do
    $cnt \leftarrow 0$
    for $j \leftarrow 1, m$ do
        if $\hat{\boldsymbol{P}}[i][j] \geq \boldsymbol{t}[j]$ then
            $cnt \leftarrow cnt + 1$
            $y^* \leftarrow j$                                ▷ guess of true label
    $\tilde{y} \leftarrow \tilde{\boldsymbol{y}}[i]$
    if $cnt > 1$ then                                    ▷ if label collision
        $y^* \leftarrow \arg\max \hat{\boldsymbol{P}}[i]$
    if $cnt > 0$ then
        $\boldsymbol{C}[\tilde{y}][y^*] \leftarrow \boldsymbol{C}[\tilde{y}][y^*] + 1$
output $\boldsymbol{C}$, the $m \times m$ unnormalized counts matrix
```

Appendix B. The confident joint and joint algorithms

The confident joint is expressed succinctly in equation Eqn. 1 with the thresholds expressed in Eqn. 2. For clarity, we provide these equations in algorithm form (See Alg. 1 and Alg. 2).

The confident joint algorithm (Alg. 1) is an $\mathcal{O}(m^2+nm)$ step procedure to compute $\boldsymbol{C}_{\tilde{y},y^*}$. The algorithm takes two inputs: (1) $\hat{\boldsymbol{P}}$ an $n \times m$ matrix of out-of-sample predicted probabilities $\hat{\boldsymbol{P}}[i][j] \coloneqq \hat{p}(\tilde{y}=j; x_i, \boldsymbol{\theta})$ and (2) the associated array of noisy labels. We typically use cross-validation to compute $\hat{\boldsymbol{P}}$ for train sets and a model trained on the train set and fine-tuned with cross-validation on the test set to compute $\hat{\boldsymbol{P}}$ for a test set. Any method works as long $\hat{p}(\tilde{y}=j; \boldsymbol{x}, \boldsymbol{\theta})$ are out-of-sample, holdout predicted probabilities.

Computation time. Finding label errors in ImageNet takes 3 minutes on an i7 CPU. Results in all tables reproducible via open-sourced `cleanlab` package.

Note that Alg. 1 embodies Eqn. 1, and Alg. 2 realizes Eqn. 3.

Algorithm 2 (**Joint**) calibrates the confident joint to estimate the latent, true distribution of class-conditional label noise

```
  input $\boldsymbol{C}_{\tilde{y},y^*}[i][j]$, $m \times m$ unnormalized counts
  input $\tilde{\boldsymbol{y}}$ an $n \times 1$ array of noisy integer labels
  procedure JointEstimation($\boldsymbol{C}$, $\tilde{\boldsymbol{y}}$):
$\tilde{\boldsymbol{C}}_{\tilde{y}=i,y^*=j} \leftarrow \frac{\boldsymbol{C}_{\tilde{y}=i,y^*=j}}{\sum_{j\in[m]} \boldsymbol{C}_{\tilde{y}=i,y^*=j}} \cdot |\boldsymbol{X}_{\tilde{y}=i}|$          ▷ calibrate marginals
$\hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j} \leftarrow \frac{\tilde{\boldsymbol{C}}_{\tilde{y}=i,y^*=j}}{\sum\limits_{i\in[m],j\in[m]} \tilde{\boldsymbol{C}}_{\tilde{y}=i,y^*=j}}$          ▷ joint sums to 1
  output $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ joint dist. matrix $\sim p(\tilde{y}, y^*)$
```

Appendix C. Extended Comparison of Confident Learning Methods on CIFAR-10

Fig. S1 shows the absolute difference of the true joint $\boldsymbol{Q}_{\tilde{y},y^*}$ and the joint distribution estimated using confident learning $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ on CIFAR-10, for 20%, 40%, and 70% label noise, 20%, 40%, and 60% sparsity, for all pairs of classes in the joint distribution of label noise. Observe that in moderate noise regimes between 20% and 40% noise, confident learning accurately estimates nearly every entry in the joint distribution of label noise. This figure serves to provide evidence for how confident learning identifies the label errors with high accuracy as shown in Table 2 as well as support our theoretical contribution that confident learning exactly estimates the joint distribution of labels under reasonable assumptions (c.f., Thm. 2).

Because we did not remove label errors from the validation set, when training on the data cleaned by CL in the train set, we may have induced a distributional shift, making the moderate increase accuracy a more satisfying result.

In Table S1, we estimate the $\boldsymbol{Q}_{\tilde{y},y^*}$ using the confusion-matrix $\boldsymbol{C}_{\text{confusion}}$ approach normalized via Eqn. (3) and compare this $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$, estimated by normalizing the CL approach with the confident joint $\boldsymbol{C}_{\tilde{y},y^*}$, for various amounts of noise and sparsity in $\boldsymbol{Q}_{\tilde{y},y^*}$. Table S1 shows improvement using $\boldsymbol{C}_{\tilde{y},y^*}$ over $\boldsymbol{C}_{\text{confusion}}$, low RMSE scores, and robustness to sparsity in moderate-noise regimes.

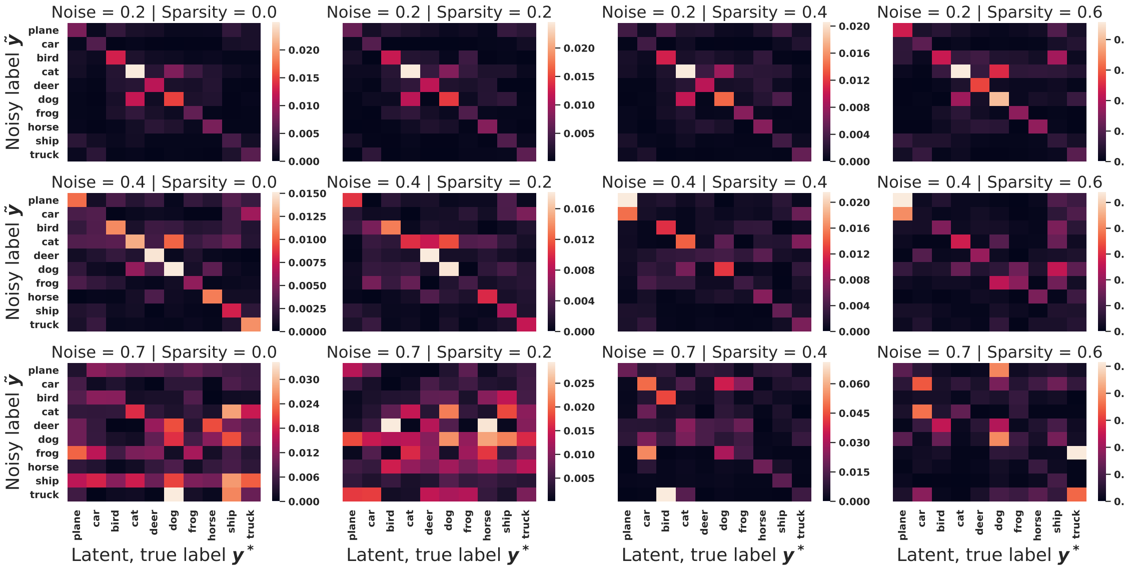


Figure S1: Absolute difference of the true joint $\boldsymbol{Q}_{\tilde{y},y^*}$ and the joint distribution estimated using confident learning $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ on CIFAR-10, for 20%, 40%, and 70% label noise, 20%, 40%, and 60% sparsity, for all pairs of classes in the joint distribution of label noise.

Table S1: RMSE error of $\boldsymbol{Q}_{\tilde{y},y^*}$ estimation on CIFAR-10 using $\boldsymbol{C}_{\tilde{y},y^*}$ to estimate $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ compared with using the baseline approach $\boldsymbol{C}_{\text{confusion}}$ to estimate $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$.

Noise	0.2				0.4				0.7			
Sparsity	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6
$\lVert\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ - $\boldsymbol{Q}_{\tilde{y},y^*}\rVert_2$	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.004**	**0.005**	0.011	**0.010**	0.015	**0.017**
$\lVert\hat{\boldsymbol{Q}}_{confusion}$ - $\boldsymbol{Q}_{\tilde{y},y^*}\rVert_2$	0.006	0.006	0.005	0.005	0.005	0.005	0.005	0.007	0.011	0.011	0.015	0.019

C.1 Benchmarking INCV

We benchmarked INCV using the official Github code[2] on a machine with 128 GB of RAM and 4 RTX 2080 ti GPUs. Due to memory leak issues (as of the February 2020 open-source release, tested on a MacOS laptop with 16GB RAM and Ubuntu 18.04 LTS Linux server 128GB RAM) in the implementation, training frequently stopped due to out-of-memory errors. For fair comparison, we restarted INCV training until all models completed at least 90 training epochs. For each experiment, Table S2 shows the total time required for training, epochs completed, and the associated accuracies. As shown in the table, the training time for INCV may take over 20 hours because the approach requires iterative retraining. For comparison, CL takes less than three hours on the same machine: an hour for cross-validation, less than a minute to find errors, an hour to retrain.

[2] https://github.com/chenpf1025/noisy_label_understanding_utilizing

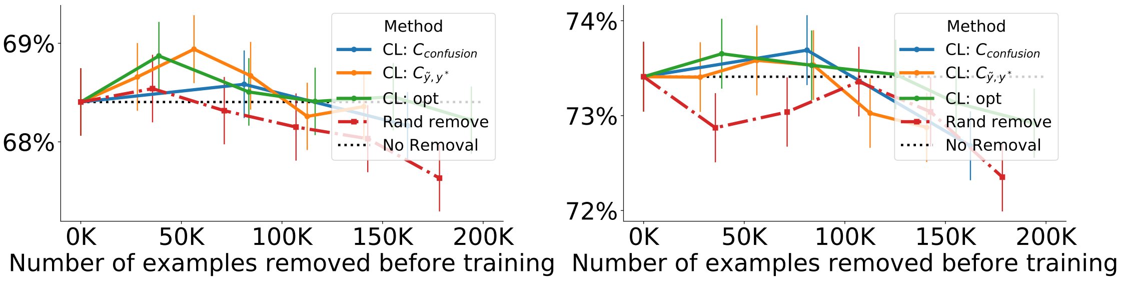
Table S2: Information about INCV benchmarks including accuracy, time, and epochs trained for various noise and sparsity settings.

Noise	0.2				0.4				0.7			
Sparsity	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6
Accuracy	0.878	0.886	0.896	0.892	0.844	0.766	0.854	0.736	0.283	0.253	0.348	0.297
Time (hours)	9.120	11.350	10.420	7.220	7.580	11.720	20.420	6.180	16.230	17.250	16.880	18.300
Epochs trained	91	91	200	157	91	200	200	139	92	92	118	200

Appendix D. Additional Figures

In this section, we include additional figures that support the main manuscript. Fig. S2 explores the benchmark accuracy of the individual confident learning approaches to support Fig. 5 and Fig. 4 in the main text. The noise matrices shown in Fig. S3 were used to generate the synthetic noisy labels for the results in Tables 4 and 2.

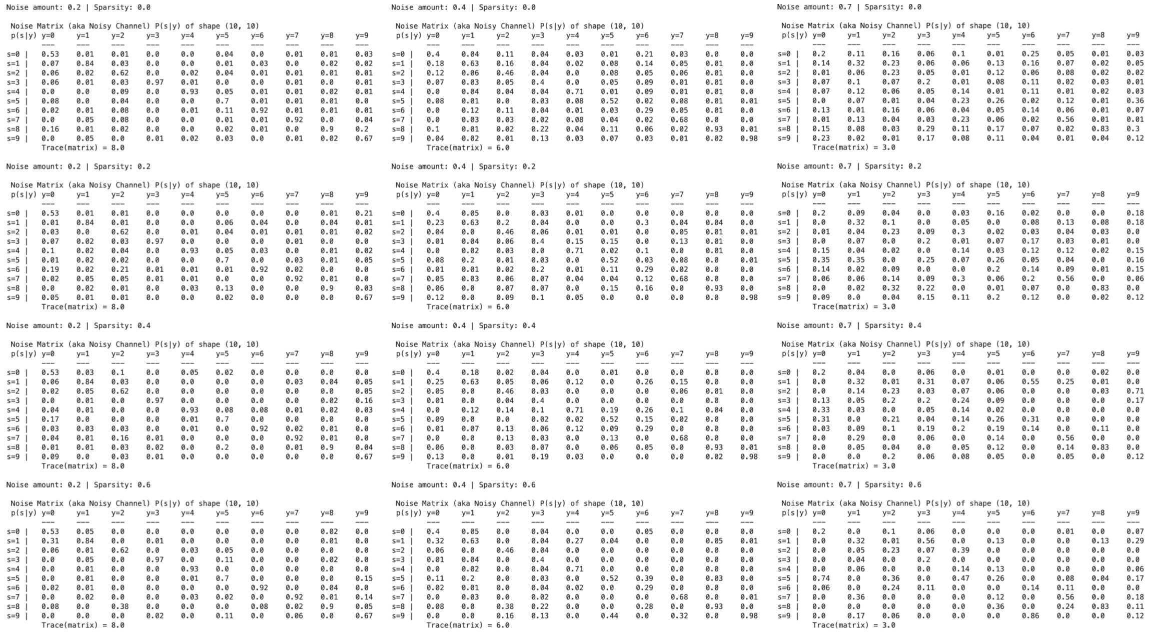
Fig. S2 shows the top-1 accuracy on the ILSVRC validation set when removing label errors estimated by CL methods versus removing random examples. For each CL method, we plot the accuracy of training with 20%, 40%,..., 100% of the estimated label errors removed, omitting points beyond 200k.



(a) ResNet18 Validation Accuracy

(b) ResNet50 Validation Accuracy

Figure S2: Increased ResNet validation accuracy using CL methods on ImageNet with original labels (no synthetic noise added). Each point on the line for each method, from left to right, depicts the accuracy of training with 20%, 40%..., 100% of estimated label errors removed. Error bars are estimated with Clopper-Pearson 95% confidence intervals. The red dash-dotted baseline captures when examples are removed uniformly randomly. The black dotted line depicts accuracy when training with all examples.



```
Noise amount: 0.2 | Sparsity: 0.0

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.53   0.01   0.01   0.0    0.0    0.04   0.0    0.01   0.01   0.03
s=1 |   0.07   0.84   0.03   0.0    0.0    0.01   0.03   0.0    0.02   0.02
s=2 |   0.06   0.02   0.62   0.0    0.02   0.04   0.01   0.01   0.01   0.01
s=3 |   0.06   0.01   0.03   0.97   0.01   0.0    0.0    0.01   0.0    0.01
s=4 |   0.0    0.0    0.09   0.0    0.93   0.05   0.01   0.01   0.02   0.01
s=5 |   0.08   0.0    0.04   0.0    0.0    0.7    0.01   0.01   0.01   0.0
s=6 |   0.02   0.01   0.08   0.0    0.01   0.11   0.92   0.01   0.01   0.01
s=7 |   0.0    0.05   0.08   0.0    0.0    0.01   0.01   0.92   0.0    0.04
s=8 |   0.16   0.01   0.02   0.0    0.0    0.02   0.01   0.0    0.9    0.2
s=9 |   0.0    0.05   0.0    0.01   0.02   0.03   0.0    0.01   0.02   0.67
        Trace(matrix) = 8.0
```

```
Noise amount: 0.4 | Sparsity: 0.0

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.4    0.04   0.11   0.04   0.03   0.01   0.21   0.03   0.0    0.0
s=1 |   0.18   0.63   0.16   0.04   0.02   0.08   0.14   0.05   0.01   0.0
s=2 |   0.12   0.06   0.46   0.04   0.0    0.08   0.05   0.06   0.01   0.0
s=3 |   0.07   0.03   0.05   0.4    0.0    0.05   0.09   0.01   0.01   0.0
s=4 |   0.0    0.04   0.04   0.04   0.71   0.01   0.09   0.01   0.01   0.0
s=5 |   0.08   0.01   0.0    0.03   0.08   0.52   0.02   0.08   0.01   0.01
s=6 |   0.0    0.12   0.11   0.04   0.01   0.03   0.29   0.05   0.01   0.0
s=7 |   0.0    0.03   0.03   0.02   0.08   0.04   0.02   0.68   0.0    0.0
s=8 |   0.1    0.01   0.02   0.22   0.04   0.11   0.06   0.02   0.93   0.01
s=9 |   0.04   0.02   0.01   0.13   0.03   0.07   0.03   0.01   0.02   0.98
        Trace(matrix) = 6.0
```

```
Noise amount: 0.7 | Sparsity: 0.0

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.2    0.11   0.16   0.06   0.1    0.01   0.25   0.05   0.01   0.03
s=1 |   0.14   0.32   0.23   0.06   0.06   0.13   0.16   0.07   0.02   0.05
s=2 |   0.01   0.06   0.23   0.05   0.01   0.12   0.06   0.08   0.02   0.02
s=3 |   0.07   0.1    0.07   0.2    0.01   0.08   0.11   0.02   0.03   0.01
s=4 |   0.07   0.12   0.06   0.05   0.14   0.01   0.11   0.01   0.02   0.03
s=5 |   0.0    0.07   0.01   0.04   0.23   0.26   0.02   0.12   0.01   0.36
s=6 |   0.13   0.01   0.16   0.06   0.04   0.05   0.14   0.06   0.01   0.07
s=7 |   0.01   0.13   0.04   0.03   0.23   0.06   0.02   0.56   0.01   0.01
s=8 |   0.15   0.08   0.03   0.29   0.11   0.17   0.07   0.02   0.83   0.3
s=9 |   0.23   0.02   0.01   0.17   0.08   0.11   0.04   0.01   0.04   0.12
        Trace(matrix) = 3.0
```

```
Noise amount: 0.2 | Sparsity: 0.2

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.53   0.01   0.01   0.0    0.0    0.0    0.0    0.0    0.01   0.21
s=1 |   0.01   0.84   0.01   0.0    0.0    0.06   0.04   0.0    0.04   0.01
s=2 |   0.03   0.0    0.62   0.0    0.01   0.04   0.01   0.01   0.01   0.02
s=3 |   0.07   0.02   0.03   0.97   0.0    0.0    0.0    0.01   0.0    0.0
s=4 |   0.1    0.02   0.04   0.0    0.93   0.05   0.03   0.0    0.01   0.02
s=5 |   0.01   0.02   0.02   0.0    0.0    0.7    0.0    0.03   0.01   0.05
s=6 |   0.19   0.02   0.21   0.01   0.01   0.01   0.92   0.02   0.0    0.0
s=7 |   0.02   0.05   0.05   0.01   0.01   0.0    0.0    0.92   0.01   0.0
s=8 |   0.0    0.02   0.01   0.0    0.03   0.13   0.0    0.0    0.9    0.03
s=9 |   0.05   0.01   0.01   0.0    0.0    0.02   0.0    0.0    0.0    0.67
        Trace(matrix) = 8.0
```

```
Noise amount: 0.4 | Sparsity: 0.2

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.4    0.05   0.0    0.03   0.01   0.0    0.0    0.0    0.0    0.0
s=1 |   0.23   0.63   0.2    0.04   0.0    0.0    0.3    0.04   0.04   0.0
s=2 |   0.04   0.0    0.46   0.06   0.01   0.01   0.0    0.05   0.01   0.01
s=3 |   0.01   0.04   0.06   0.4    0.15   0.15   0.0    0.13   0.01   0.0
s=4 |   0.0    0.02   0.03   0.0    0.71   0.02   0.1    0.0    0.01   0.0
s=5 |   0.08   0.2    0.01   0.03   0.0    0.52   0.03   0.08   0.0    0.01
s=6 |   0.01   0.01   0.02   0.2    0.01   0.11   0.29   0.02   0.0    0.0
s=7 |   0.05   0.03   0.06   0.07   0.04   0.04   0.12   0.68   0.0    0.0
s=8 |   0.06   0.0    0.07   0.07   0.0    0.15   0.16   0.0    0.93   0.0
s=9 |   0.12   0.0    0.09   0.1    0.05   0.0    0.0    0.0    0.0    0.98
        Trace(matrix) = 6.0
```

```
Noise amount: 0.7 | Sparsity: 0.2

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.2    0.09   0.04   0.0    0.03   0.16   0.02   0.0    0.0    0.18
s=1 |   0.0    0.32   0.1    0.0    0.05   0.0    0.08   0.13   0.08   0.18
s=2 |   0.01   0.04   0.23   0.09   0.3    0.02   0.03   0.04   0.03   0.0
s=3 |   0.0    0.07   0.0    0.2    0.01   0.07   0.17   0.03   0.01   0.0
s=4 |   0.15   0.04   0.02   0.0    0.14   0.03   0.12   0.12   0.02   0.15
s=5 |   0.35   0.35   0.0    0.25   0.07   0.26   0.05   0.04   0.0    0.16
s=6 |   0.14   0.02   0.09   0.0    0.0    0.2    0.14   0.09   0.01   0.15
s=7 |   0.06   0.06   0.14   0.09   0.3    0.06   0.2    0.56   0.0    0.06
s=8 |   0.0    0.02   0.32   0.22   0.0    0.01   0.07   0.0    0.83   0.0
s=9 |   0.09   0.0    0.04   0.15   0.11   0.2    0.12   0.0    0.02   0.12
        Trace(matrix) = 3.0
```

```
Noise amount: 0.2 | Sparsity: 0.4

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.53   0.03   0.1    0.0    0.05   0.02   0.0    0.0    0.0    0.0
s=1 |   0.06   0.84   0.03   0.0    0.0    0.0    0.0    0.03   0.04   0.05
s=2 |   0.02   0.05   0.62   0.0    0.0    0.0    0.0    0.0    0.0    0.05
s=3 |   0.0    0.01   0.0    0.97   0.0    0.0    0.0    0.0    0.02   0.16
s=4 |   0.04   0.01   0.0    0.0    0.93   0.08   0.08   0.01   0.02   0.03
s=5 |   0.17   0.0    0.0    0.0    0.01   0.7    0.0    0.0    0.0    0.0
s=6 |   0.03   0.03   0.03   0.0    0.01   0.0    0.92   0.02   0.01   0.0
s=7 |   0.04   0.01   0.16   0.01   0.0    0.0    0.0    0.92   0.01   0.0
s=8 |   0.01   0.01   0.03   0.02   0.0    0.2    0.0    0.01   0.9    0.04
s=9 |   0.09   0.0    0.03   0.01   0.0    0.0    0.0    0.0    0.0    0.67
        Trace(matrix) = 8.0
```

```
Noise amount: 0.4 | Sparsity: 0.4

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.4    0.18   0.02   0.04   0.0    0.01   0.0    0.0    0.0    0.0
s=1 |   0.25   0.63   0.05   0.06   0.12   0.0    0.26   0.15   0.0    0.0
s=2 |   0.05   0.0    0.46   0.03   0.0    0.0    0.0    0.06   0.01   0.0
s=3 |   0.01   0.0    0.04   0.4    0.0    0.0    0.0    0.0    0.0    0.0
s=4 |   0.0    0.12   0.14   0.1    0.71   0.19   0.26   0.1    0.04   0.0
s=5 |   0.09   0.0    0.0    0.02   0.02   0.52   0.15   0.02   0.0    0.0
s=6 |   0.01   0.07   0.13   0.06   0.12   0.09   0.29   0.0    0.0    0.0
s=7 |   0.0    0.0    0.13   0.03   0.0    0.13   0.0    0.68   0.0    0.0
s=8 |   0.06   0.0    0.03   0.07   0.0    0.06   0.05   0.0    0.93   0.01
s=9 |   0.13   0.0    0.01   0.19   0.03   0.0    0.0    0.0    0.02   0.98
        Trace(matrix) = 6.0
```

```
Noise amount: 0.7 | Sparsity: 0.4

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.2    0.04   0.0    0.06   0.0    0.01   0.0    0.0    0.02   0.0
s=1 |   0.0    0.32   0.01   0.31   0.07   0.06   0.55   0.25   0.01   0.0
s=2 |   0.0    0.14   0.23   0.03   0.07   0.06   0.0    0.0    0.03   0.71
s=3 |   0.13   0.05   0.2    0.2    0.24   0.09   0.0    0.0    0.0    0.17
s=4 |   0.33   0.03   0.0    0.05   0.14   0.02   0.0    0.0    0.0    0.0
s=5 |   0.31   0.0    0.21   0.04   0.14   0.26   0.31   0.0    0.0    0.0
s=6 |   0.03   0.09   0.1    0.19   0.2    0.19   0.14   0.0    0.11   0.0
s=7 |   0.0    0.29   0.0    0.06   0.0    0.14   0.0    0.56   0.0    0.0
s=8 |   0.0    0.05   0.04   0.0    0.05   0.12   0.0    0.14   0.83   0.0
s=9 |   0.0    0.0    0.2    0.06   0.08   0.05   0.0    0.05   0.0    0.12
        Trace(matrix) = 3.0
```

```
Noise amount: 0.2 | Sparsity: 0.6

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.53   0.05   0.0    0.0    0.0    0.0    0.0    0.0    0.02   0.0
s=1 |   0.31   0.84   0.0    0.01   0.0    0.0    0.0    0.0    0.01   0.0
s=2 |   0.06   0.01   0.62   0.0    0.03   0.05   0.0    0.0    0.0    0.0
s=3 |   0.0    0.05   0.0    0.97   0.0    0.11   0.0    0.0    0.02   0.0
s=4 |   0.0    0.01   0.0    0.0    0.93   0.0    0.0    0.0    0.0    0.0
s=5 |   0.0    0.01   0.0    0.0    0.01   0.7    0.0    0.0    0.0    0.15
s=6 |   0.02   0.01   0.0    0.0    0.0    0.0    0.92   0.0    0.04   0.0
s=7 |   0.0    0.02   0.0    0.0    0.03   0.02   0.0    0.92   0.01   0.14
s=8 |   0.08   0.0    0.38   0.0    0.0    0.0    0.08   0.02   0.9    0.05
s=9 |   0.0    0.0    0.0    0.02   0.0    0.11   0.0    0.06   0.0    0.67
        Trace(matrix) = 8.0
```

```
Noise amount: 0.4 | Sparsity: 0.6

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.4    0.05   0.0    0.04   0.0    0.0    0.05   0.0    0.0    0.0
s=1 |   0.32   0.63   0.0    0.04   0.27   0.04   0.0    0.0    0.05   0.01
s=2 |   0.06   0.0    0.46   0.04   0.0    0.0    0.0    0.0    0.0    0.0
s=3 |   0.01   0.04   0.0    0.4    0.0    0.0    0.0    0.0    0.0    0.0
s=4 |   0.0    0.02   0.0    0.04   0.71   0.0    0.0    0.0    0.0    0.0
s=5 |   0.11   0.2    0.0    0.03   0.0    0.52   0.39   0.0    0.03   0.0
s=6 |   0.02   0.01   0.0    0.04   0.02   0.0    0.29   0.0    0.0    0.0
s=7 |   0.0    0.03   0.0    0.02   0.0    0.0    0.0    0.68   0.0    0.01
s=8 |   0.08   0.0    0.38   0.22   0.0    0.0    0.28   0.0    0.93   0.0
s=9 |   0.0    0.0    0.16   0.13   0.0    0.44   0.0    0.32   0.0    0.98
        Trace(matrix) = 6.0
```

```
Noise amount: 0.7 | Sparsity: 0.6

 Noise Matrix (aka Noisy Channel) P(s|y) of shape (10, 10)
 p(s|y) y=0    y=1    y=2    y=3    y=4    y=5    y=6    y=7    y=8    y=9
        ---    ---    ---    ---    ---    ---    ---    ---    ---    ---
s=0 |   0.2    0.0    0.1    0.06   0.0    0.0    0.0    0.01   0.0    0.07
s=1 |   0.0    0.32   0.01   0.56   0.0    0.13   0.0    0.0    0.13   0.29
s=2 |   0.0    0.05   0.23   0.07   0.39   0.0    0.0    0.0    0.0    0.0
s=3 |   0.0    0.04   0.0    0.2    0.0    0.0    0.0    0.0    0.0    0.0
s=4 |   0.0    0.06   0.0    0.0    0.14   0.13   0.0    0.0    0.0    0.06
s=5 |   0.74   0.0    0.36   0.0    0.47   0.26   0.0    0.08   0.04   0.17
s=6 |   0.06   0.0    0.24   0.11   0.0    0.0    0.14   0.11   0.0    0.0
s=7 |   0.0    0.36   0.0    0.0    0.0    0.12   0.0    0.56   0.0    0.18
s=8 |   0.0    0.0    0.0    0.0    0.0    0.36   0.0    0.24   0.83   0.11
s=9 |   0.0    0.17   0.06   0.0    0.0    0.0    0.86   0.0    0.0    0.12
        Trace(matrix) = 3.0
```

Figure S3: The CIFAR-10 noise transition matrices used to create the synthetic label errors. In the cleanlab code base, s is used in place of $\tilde{y}$ to notate the noisy unobserved labels and y is used in place of y^* to notate the latent uncorrupted labels.