TrustNet: Learning from Trusted Data Against (A)symmetric Label Noise

Amirmasoud Ghiassi
Delft University of Technology
Delft, The Netherlands
s.ghiassi@tudelft.nl

Robert Birke
ABB Research
Baden-Dättwil, Switzerland
robert.birke@ch.abb.com

Lydia Y.Chen
Delft University of Technology
Delft, The Netherlands
lydiaychen@ieee.org

ABSTRACT

Big Data systems allow collecting massive datasets to feed the data hungry deep learning. Labelling these ever-bigger datasets is increasingly challenging and label errors affect even highly curated sets. This makes robustness to label noise a critical property for weakly-supervised classifiers. The related works on resilient deep networks tend to focus on a limited set of synthetic noise patterns, and with disparate views on their impacts, e.g., robustness against symmetric v.s. asymmetric noise patterns. In this paper, we first extend the theoretical analysis of test accuracy for any given noise patterns. Based on the insights, we design TrustNet that first learns the pattern of noise corruption, being it both symmetric or asymmetric, from a small set of trusted data. Then, TrustNet is trained via a robust loss function, which weights the given labels against the inferred labels from the learned noise pattern. The weight is adjusted based on model uncertainty across training epochs. We evaluate TrustNet on synthetic label noise for CIFAR-10, CIFAR-100 and big real-world data with label noise, i.e., Clothing1M. We compare against state-of-the-art methods demonstrating the strong robustness of TrustNet under a diverse set of noise patterns.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Machine learning approaches** → *Neural networks.*

KEYWORDS

deep neural networks, robust loss function, noisy labels in big data, noise transition matrix, noise estimation

ACM Reference Format:
Amirmasoud Ghiassi, Robert Birke, and Lydia Y.Chen. 2021. TrustNet: Learning from Trusted Data Against (A)symmetric Label Noise. In *2021 IEEE/ACM 7th International Conference on Big Data Computing, Applications and Technologies (BDCAT '21) (BDCAT '21), December 6–9, 2021, Leicester, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3492324.3494166

1 INTRODUCTION

Nowadays big data systems allow collecting and processing immense datasets which shifts the bottleneck for deep learning from computing resources to providing high quality labels [26]. Big data systems allowed for a surge of massive self-generated data. However, it is shown that big data training sets collected from the wild can contain corrupted labels as high as 40% [38]. Even popular and curated learning datasets include varying degrees of wrong labels with bigger sets tending to have higher noise ratios, e.g., 10.12% for the QuickDraw dataset with 50M samples and 5.83% for the ImageNet dataset with 50K samples [23, 24]. The high learning capacity of deep neural networks can memorize the pattern of correct data and, unfortunately, dirty data as well [1]. As a result, when training on data with non-negligible dirty labels, the learning accuracy of deep neural networks can significantly drop [42].

While the prior art deems it imperative to derive robust neural networks that are resilient to label noise, there is a disparity in which noise patterns to consider and evaluate. The majority of deep networks robust against dirty labels focuses on synthetic label noise, which can be symmetric or asymmetric. The former case [2] assumes noise labels can be corrupted into any other classes with equal probability. The later case [36] assumes only a particular set of classes are swapped, e.g., truck images are often mislabeled as automobile class in CIFAR-10. Patterns of noisy labels observed from real-life big data sets, e.g., Clothing1M [38], exhibit not only high percentages of label noise but also more complicated patterns mixing symmetric and asymmetric noises. Moreover, there is disagreement among related work on which noise patterns are more detrimental and difficult to defend against for regular networks [21, 34].

Noise patterns are commonly captured in transition matrices [2], which describe the probability of how a true label is corrupted into another fake and observable label. A large body of prior art estimates such a label transition matrix without knowing the true labels and incorporates such information into the learning process [25]. Accurate estimation of the transition matrix can improve the robustness of neural networks, but it is extremely complicated when lacking the information on true labels and encountering sophisticated noise patterns [39, 40].

Joint training on clean and adversarial examples with known ground truth is shown effective [15, 20] to enhance the robustness of deep models against noisy and poisonous labels. Nonetheless, it is costly to obtain label ground truth. To take advantage of adversarial examples and avoid its high overhead, we advocate to use only a fraction of *trusted data* that contain not only given labels but also the expert-validated true labels for the same. Moreover, we opt to use such small set to mainly supervise the training of transition matrix, instead of supervising the classifier directly as done in most adversarial learning.

In this paper, we first develop a thorough understanding of the noise patterns, ranging from symmetric and asymmetric. We extend

BDCAT '21, December 6–9, 2021, Leicester, United Kingdom
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9164-1/21/12.
https://doi.org/10.1145/3492324.3494166

the analysis from [2] and derive the generalized analysis for classification test accuracy under any given noise pattern. Our theoretical analysis compares real-world noise patterns against synthetic, symmetric, and simple asymmetric, noise. Our findings on a diverse set of noise patterns lead us to focus on challenging cases where existing robust networks [6, 25, 37] may fall short of defending against.

The second contribution of this paper is to introduce a new robust learning framework TrustNet. Specifically, we adopt the idea in LABELNET [5] to estimate the noise transition matrix via training on a small set of trusted data, i.e., 10% of the training data and provide estimated labels – additional label information. TrustNet extends LABELNET by weighting the loss of the given labels and inferred labels to enhance the model performance. The specific weights are dynamically adjusted every epoch, based on the model confidence.

Thirdly, we evaluate TrustNet on multiple big data vision sets. We use the curated CIFAR-10 and CIFAR-100 sets with labels corrupted by synthetically generated noise transition patterns. TrustNet is able to achieve higher accuracy than SCL [36], D2L [37], Boostrap [27], Forward [25], and Co-teaching+ [41] in all most challenging scenarios. We also demonstrate the effectiveness of TrustNet on a big data vision set collected in the wild, i.e., Clothing1M, again achieving higher accuracy than state-of-the-art baselines.

2 RELATED WORK

The problem of noisy labeled data has been addressed in several recent studies. We first summarize the impact of noise patterns, followed by the defense strategies that specifically leverage noise patterns.

2.1 Noisy Labels in Big Data

The existence of wrong labels in Big Data sets is inevitable [24]. Several studies indicate the presence of noisy labels in both training sets [3, 4, 13] and testing sets [23, 24, 29]. The amount of errors, i.e. noise level, varies according to the label collection method, the annotators expertise, and, most relevantly, the size of the dataset [33]. For instance, [19] and [13, 30] study the noisy labels in the WebVision and ImageNet datasets, respectively, two of popular big vision datasets with over 24.M and 14M images. However, this phenomena goes beyond image labelling. Recent studies [23, 24] find label errors in many even highly popular learning datasets from diverse domains.

2.2 Impact of Noise Patterns

Understanding the effect of label noise on the performance of the learning models is crucial to make them robust. The impact of label noise in deep neural networks is first characterized [2] by the theoretical testing accuracy over a limited set of noise patterns. We generalize the theoretical test accuracy proposed by [2] for different noise patterns by using a generic transition matrix. [34] suggests an undirected graphical model for modeling label noise in deep neural networks, indicating the symmetric noise to be more challenging than asymmetric. Multiple untrusted data sources are studied by [16], considering label noise as one of the attributes of mistrust. However, it remains unclear how various kinds of noise patterns impact learning.

2.3 Noise Resilient Networks

2.3.1 *Symmetric Noise.*

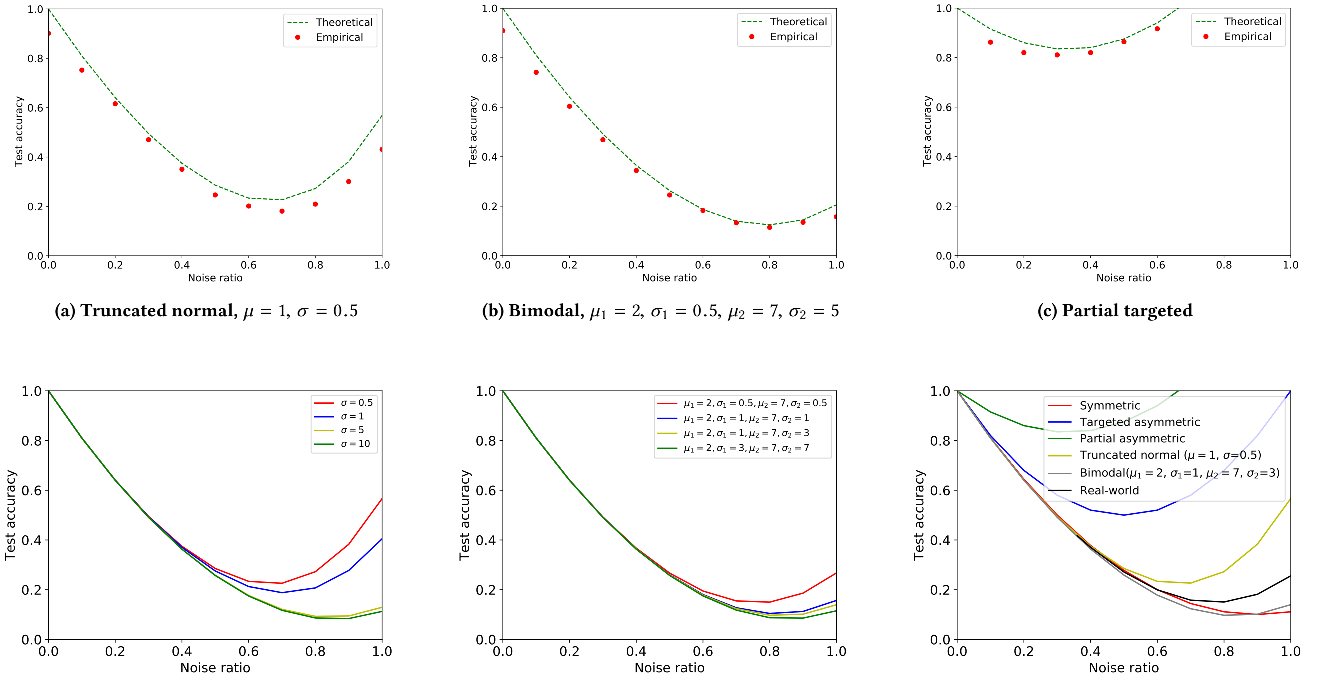
The following studies tackle the problem of symmetric label noise, meaning that corrupted labels can be any of the remaining classes with equal probability. One approach is to train the network based on noise resilient loss functions. D2L [21] monitors the changes in Local Intrinsic Dimension (LID) and incorporates LID into their loss function for the symmetric label noise. [12] introduces a loss correction technique and estimates a label corruption matrix for symmetric and asymmetric noise. Leveraging two different neural networks is another method to overcome label noise. Co-teaching [10] and Co-teaching+ [41] trains two neural networks while crossing the samples with the smallest loss between the networks for both noise patterns. [14] combats uniform label flipping via a curriculum provided by the MentorNet for the StudentNet. However, these works do not explicitly model the noise pattern in their resilient models. Although LABELNET [5] learns the noise pattern by training a DNN with ground truth and noisy labels, it requires the ground truth of all the samples. We aim to solve this issue by reducing the dependency on the ground truth via TrustNet.

2.3.2 *Asymmetric Noise.*

Another stream of related work considers both symmetric and asymmetric noise. One key idea is to differentiate clean and noisy samples by exploring their dissimilarity. [11, 18] introduce class prototypes for each class and compare the samples with the prototypes to detect noisy and clean samples. Decoupling [22] uses two neural networks and updates the networks when a disagreement happens between the networks. Estimation of the noise transition matrix is another line of research to overcome label noise, introduced in Masking [9] and Forward [25] to correct the labels. However, these studies fail to consider the information in the noisy labels to estimate the matrix. Building a robust loss function against label noise has been studied in the following works, although the dynamics of the learning model seem to be neglected. SCL [36] and [43] provide robust loss function by adding regularization term. Bootstrapping [27] combines perceptual consistency with the prediction objective by using a reconstruction loss. Meta-Weight-Net [31] uses multi-layer perceptron to re-weight samples during learning process in the loss function. With the same perspective, [28] re-weights samples based on their similarity to a clean validation set. The studies [7, 32] changes the architecture of the neural network to tackle the problem. In this work, we study both symmetric and various kinds of asymmetric label noise. We leverage the information of the trusted data, containing both noisy labels and ground truth, to accurately estimate the noise transition matrix. Furthermore, we benefit from a dynamic update in our proposed loss function to tackle the label noise problem.

3 UNDERSTANDING DNNS TRAINED WITH NOISY LABELS

In this section, we present theoretical analysis on the test accuracy of deep neural networks assumed to have high learning capacity. Test accuracy is a common metric defined as the probability that the predicted label is equal to the given label. We extend prior



(a) Truncated normal, $\mu = 1$, $\sigma = 0.5$

(b) Bimodal, $\mu_1 = 2$, $\sigma_1 = 0.5$, $\mu_2 = 7$, $\sigma_2 = 5$

(c) Partial targeted

(d) Truncated normal: $\mu = 1$, different σ

(e) Bimodal: $\mu_1 = 2$, $\mu_2 = 7$, different σ_1, σ_2

(f) All

Figure 1: Three study cases on CIFAR-10 with 10 classes. First row compares the theoretical values (lines) against empirical test accuracy results (points) for $0 \leq \varepsilon \leq 1$. Second row shows analytical results: the impact of noise patterns with different parameters on the test accuracy across noise ratios.

art results [2] by deriving test accuracy for generic label noise distributions. We apply our formulation on three exemplary study cases and verify the theoretical values against experimental results. Finally, we compare test accuracy curves for different noise patterns providing insights on their difficulty for regular networks.

3.1 Preliminaries

Consider the classification problem having dataset $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ..., (\boldsymbol{x}_N, y_N)\}$ where $\boldsymbol{x}_k$ denotes the k^{th} observed sample, and $y_k \in C := \{0, ..., c-1\}$ the corresponding given class label over c classes affected by label noise. Let $\mathcal{F}(\cdot, \boldsymbol{\theta})$ denote a neural network parameterized by $\boldsymbol{\theta}$, and $y^{\mathcal{F}}$ denote the predicted label of $\boldsymbol{x}$ given by the network $y^{\mathcal{F}} = \mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})$. The label corruption process is characterised by a transition matrix $T_{ij} = P(y = j | \hat{y} = i)$ where $\hat{y}$ is the true label. Synthetic noise patterns are expressed as a label corruption probability ε plus a noise label distribution. For example, symmetric noise is defined by ε describing the corruption probability, i.e. $T_{ii} = 1 - \varepsilon, \forall i \in C$, plus a uniform label distribution across the other labels, i.e. $T_{ij} = \frac{\varepsilon}{c-1}, \forall i \neq j \in C$.

3.2 Generalization of Test Accuracy

To generalize the previous test accuracy [2], we first consider the case where all classes are affected by the same noise ratio. We then further extend to the case where only a subset of classes is affected by noise. To derive the following Lemmas we assume that $\mathcal{F}$ is a perfect Deep Neural Network (DNN) having sufficient high capacity to learn the given pattern with high accuracy. This is the same assumption used by related work, i.e. [2, 42].

All class noise: All classes are affected by the same noise ratio ε, i.e., meaning only $1 - \varepsilon$ percentage of given labels are the true labels.

Lemma 1. *For noise with fixed noise ratio ε and any given label distribution with probability function $P(y = j), \forall j \neq i$, where $i \in C$ is the true label, the test accuracy is*

$$P(y^{\mathcal{F}} = y) = (1 - \varepsilon)^2 + \varepsilon^2 \sum_{j \neq i}^{C} P^2(y = j) \tag{1}$$

The proof is available in the appendix.

Partial class noise: in this pattern only a subset S of class labels are affected by a noise ratio, whereas the set $U = C \setminus S$ is unaffected by any label noise.

Lemma 2. *For partial class noise with equal class label probability, where S is the set affected by noise with ratio ε and U is the set of unaffected labels, for any true label $i \in C$ and any given label distribution with probability function $P(y = j), \forall j \neq i$, the test accuracy* *is*

$$P(y^{\mathcal{F}} = y) = \frac{|U|}{|C|} + \frac{|S|}{|C|}[(1-\varepsilon)^2 + \varepsilon^2 \sum_{j \neq i}^{S} P^2(y = j)] \qquad (2)$$

The proof is available in the appendix.

The goal of Lemma 1 and Lemma 2 is to generalize the test accuracy proposed by [2] to noises characterized by a generic transition matrix T_{ij}.

3.3 Validation of Theoretical Analysis

We validate our extension of test accuracy on three various noise patterns for CIFAR-10 under different noise ratios and comparing the theoretical estimation with empirical accuracy results.

As the first new noise pattern, we consider noisy class labels following a truncated normal distribution $\mathcal{N}^T(\mu, \sigma, a, b)$. This noise pattern is motivated by the targeted adversarial attacks [8]. We scale $\mathcal{N}^T(\mu, \sigma, a, b)$ by the number of classes and center it around a target class $\tilde{c}$ by setting $\mu = \tilde{c}$ and use σ to control how spread out the noise is. a and b simply define the class label boundaries, i.e. $a = 0$ and $b = c - 1$. To compute the test accuracy, we estimate the empirical distribution at the different classes and apply Eq. 1. The second noise pattern extends our previous case. This distribution, referred in short as bimodal hereon, combines two truncated normal distributions. It has two peaks in μ_1 and μ_2 with two different shapes controlled by σ_1 and σ_2. The peaks are centered on two different target classes $\mu_1 = \tilde{c_1}$ and $\mu_2 = \tilde{c_2}$. The third noise pattern considers partial targeted noise where only a subset of classes, $[2, 3, 4, 5, 9]$ in our example, are affected by targeted noise, i.e. swapped with a specific other class. Here we rely on Eq. 2 to estimate test accuracy. This noise pattern has been studied in [36].

Figure. 1 summarizes the results. The first row compares the theoretical curves against the empirical results obtained by corrupting CIFAR-10 dataset with different noise ratios from clean to fully corrupted data: $0 \leq \varepsilon \leq 1$. The highest deviation between theoretical (lines) and empirical (points) results occurs for truncated normal noise around $\varepsilon = 1.0$. Here the theoretical accuracy is 13.46% points worse than the measured accuracy. For the other two, the deviation is at most 7.69% and 6.73% (without considering $\varepsilon = 0.0$) for bimodal and partial targeted noise, respectively. Overall, the theoretical and empirical values match well across the whole range of noise ratios.

3.4 Impact of Different Noise Patterns

We conclude by using our theoretical analysis to compare the impact on test accuracy of different noise patterns. First, we consider different parameters for truncated normal and bimodal noises and finish with comparing all noise patterns from here, in [2] and the real-world noise pattern from [38].

Figure. 1, the second row shows all results. We start with truncated normal noise with a fixed target class and different σ. Higher values of σ result in a wider spread of label noise across adjacent classes as shown in Figure. 1d. Under lower noise ratios, e.g., $\varepsilon < 0.5$, the impact of varying σ is negligible, as shown by the overlapping curves. After that, we see that the most challenging cases are with high values of σ due to the wider spread of corrupted labels deviating from their true classes.Similarly to the previous analysis, for bimodal noise, we fix the target classes, i.e., μ_1 and μ_2, while varying the variances around the two peaks, i.e., σ_1 and σ_2. Overall the results are similar to truncated normal noise, but we can observe that the sensitivity to sigma is lower (see Figure. 1e) even if on average test accuracy of truncated normal is higher than bimodal noise. For instance, in case of $\varepsilon = 1.0$ the difference between $\sigma = 0.5$ and $\sigma = 1$ is 16.26% for truncated normal, but only 11.11% for bimodal. Hence, bimodal tends to be more challenging since lines for different σ are all more condensed around low values of accuracy with respect to truncated normal noise.

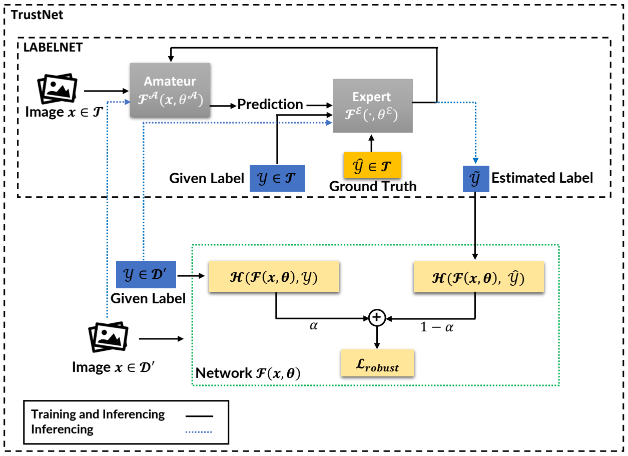

TrustNet
LABELNET
Image $x \in \mathcal{T}$
Amateur $\mathcal{F}^{\mathcal{A}}(x, \theta^{\mathcal{A}})$
Prediction
Expert $\mathcal{F}^{\mathcal{E}}(\cdot, \theta^{\mathcal{E}})$
Given Label $y \in \mathcal{T}$
$\hat{y} \in \mathcal{T}$ Ground Truth
$\tilde{y}$ Estimated Label
$y \in \mathcal{D}'$
Given Label
$\mathcal{H}(\mathcal{F}(x, \theta), y)$
$\mathcal{H}(\mathcal{F}(x, \theta), \hat{y})$
α
$1 - \alpha$
Image $x \in \mathcal{D}'$
Network $\mathcal{F}(x, \theta)$
$\mathcal{L}_{robust}$
Training and Inferencing
Inferencing

Figure 2: TrustNet architecture.

To conclude, we compare all synthetic symmetric and asymmetric noise patterns considered against the real-world noise pattern observed on the Clothing1M dataset [38] (see Figure. 1f). The measured noise ratio of this dataset is $\varepsilon = 0.41$. To estimate the test accuracy, we scale the noise pattern to different ε by redistributing the noise, such as to maintain all relative ratios between noise transition matrix elements per class. This imposes a lower limit on the noise ratio of $\varepsilon = 0.36$ to be able to keep all elements within the range $[0, 1]$. As intuition can suggest, partial targeted noise has the least impact since it only affects a fraction of classes. More interestingly, we see that the decrease in accuracy for all asymmetric noise patterns is not monotonic. When noise ratios are high, another class becomes dominant, and thus it is easier to counter the noise pattern. On the contrary, all curves tend to overlap at smaller noise ratios, i.e., noise patterns play a weaker role compared to at higher noise ratios. Finally, the real-world noise pattern almost overlaps with bimodal. This might be due that errors in Clothing1M often are between two classes sharing visual patterns [38].

4 METHODOLOGY

In this section, we present our proposed robust learning framework, TrustNet, featuring on a light weight estimation of noise patterns and a robust loss function.

4.1 TrustNet Architecture

Consider extending the classification problem from Preliminaries section with a set of trusted data, $\mathcal{T} = \{(\boldsymbol{x}_1, y_1, \hat{y}_1), (\boldsymbol{x}_2, y_2, \hat{y}_1), ..., (\boldsymbol{x}_N, y_N, \hat{y}_N)\}$. $\mathcal{T}$ is validated by experts and has for each sample $\boldsymbol{x}$ both given y and true $\hat{y}$ class labels. Hence, our classification problem comprises two types of datasets: $\mathcal{T}$ and $\mathcal{D}$, where $\mathcal{D}$ has only the given class label y. The given class labels y in both data sets are affected by the same noise pattern and noise ratio. Further, we assume that $\mathcal{T}$ is small compared to $\mathcal{D}$, i.e. $|\mathcal{T}| << |\mathcal{D}|$, due to the cost of experts' advise. Corresponding to the two datasets, TrustNet consists of two training routines highlighted by the top and bottom halves of Figure. 2.

First (top half), we adopt the architecture of LABELNET [5] and leverage the trusted dataset to learn the underlying noise transition matrix. TrustNet uses LABELNET to learn the noise transition matrix and estimate the true labels for the untrusted data. LABELNET is a deep neural network jointly trained on the given and true labels, however, it requires the ground truth of all the data. Since acquiring the ground truth of the data in real-world scenarios is extremely expensive and time consuming, we reduce this cost by modifying LABELNET via introducing a weighted loss function, which we describe in § 4.3.

Second (bottom half), the trained LABELNET is used to derive a dataset $\mathcal{D}'$ from $\mathcal{D}$ by enriching it with estimated class labels $\tilde{y}$ inferred by LABELNET (blue path). Hence $\mathcal{D}' = \{(\boldsymbol{x}_1, y_1, \tilde{y}_1), (\boldsymbol{x}_2, y_2, \tilde{y}_2), ..., (\boldsymbol{x}_N, y_N, \tilde{y}_N)\}$. Then, we train a deep neural network, $\mathcal{F}(\cdot, \boldsymbol{\theta})$, on $\mathcal{D}'$ using the proposed robust loss function from Noise Robust Loss Function section. We note that the trusted data is used only to train LABELNET, not $\mathcal{F}(\cdot, \boldsymbol{\theta})$.

As we mentioned in § 3, the proposed Lemma 1 and Lemma 2 show an extension of the DNN memorization effect on test accuracy [2, 42] to the noise patterns used to evaluate TrustNet. TrustNet intends to reduce the memorization effect for noise with a two-stage approach. The first stage corrects the noisy labels and the second stage uses a weighted loss function on the given and the corrected labels.

4.2 Estimating Noise Transition Matrix

Here we briefly describe LABELNET which is a framework that consists of two neural networks: Amateur and Expert. Amateur aims to classify images guided by the feedback from Expert. Expert acts as a supervisor who corrects the predictions of Amateur based on the ground truth. Essentially, Expert learns how to transform predicted labels to true labels, i.e., a reverse noise transition matrix.

During training, first Amateur provides for a sample $\boldsymbol{x}_k$ a prediction of the class probabilities $\boldsymbol{y}_k^{\mathcal{A}}$ to Expert. Expert uses $\boldsymbol{y}_k^{\mathcal{A}}$ concatenated with the given class label y_k to learn to predict the ground truth class label $\hat{y_k}$. In turn, the predicted label from Expert $y_k^{\mathcal{E}}$ is provided as feedback to train Amateur. In summary, training tries to minimize recursively the following two loss functions for Amateur, described by $\mathcal{F}^{\mathcal{A}}(\cdot, \boldsymbol{\theta}^{\mathcal{A}})$ and Expert, described by $\mathcal{F}^{\mathcal{E}}(\cdot, \boldsymbol{\theta}^{\mathcal{E}})$:

$$\min_{\boldsymbol{\theta}^{\mathcal{A}}} \mathcal{L}(\mathcal{F}^{\mathcal{A}}(\boldsymbol{x}_k, \boldsymbol{\theta}^{\mathcal{A}}), y_k^{\mathcal{E}}) \tag{3}$$

$$\min_{\boldsymbol{\theta}^{\mathcal{E}}} \mathcal{L}(\mathcal{F}^{\mathcal{E}}(< \boldsymbol{y}_k^{\mathcal{A}}, y_k >, \boldsymbol{\theta}^{\mathcal{E}}), \hat{y}_k) \tag{4}$$

where $< \cdot, \cdot >$ represents vector concatenation.

The trained LABELNET can estimate the true label from an image $\boldsymbol{x_k}$:

$$\tilde{y}_k = \mathcal{F}^{\mathcal{E}}(< \mathcal{F}^{\mathcal{A}}(\boldsymbol{x}_k, \boldsymbol{\theta}^{\mathcal{A}}), y_k >, \boldsymbol{\theta}^{\mathcal{E}}). \tag{5}$$

Specifically, we use the trained LABELNET to enrich and transform $\mathcal{D}$ in $\mathcal{D}'$ by incorporating for each image $\boldsymbol{x}_k$ the inferred class label $\tilde{y}_k$. Subsequently, we use $\mathcal{D}'$ to train $\mathcal{F}(\cdot, \boldsymbol{\theta})$ via the loss function robust to noise from Noise Robust Loss Function section.

4.3 Noise Robust Loss Function

The given labels are corrupted by noise. Directly training on the given labels results in highly degraded performance as the neural network is not able to easily discern between clean and corrupted labels. To make the learning more robust to noise, TrustNet proposes to modify the loss function to leverage both given labels y and inferred labels $\tilde{y}$ from LABELNET to train $\mathcal{F}(\cdot, \boldsymbol{\theta})$.

The predicted label of $\mathcal{F}(\cdot, \boldsymbol{\theta})$ is compared, e.g., via cross-entropy loss, against both the given label and inferred label. The challenge is how to combine these two loss values. Ideally, for samples for which LABELNET and $\mathcal{F}(\cdot, \boldsymbol{\theta})$ are highly accurate, the inferred label can be trusted more. On the contrary, for samples for which LABELNET and $\mathcal{F}(\cdot, \boldsymbol{\theta})$ have low accuracy, the given labels can be trusted more. Specifically, TrustNet uses a weighted average between the loss of the predicted label from $\mathcal{F}(\boldsymbol{x}_k, \boldsymbol{\theta})$ against both the given label y_k and the LABELNET's inferred label $\tilde{y}_k$ with per sample weights α_k and $(1 - \alpha_k)$ for all samples $\boldsymbol{x}_k$ in $\mathcal{D}'$. Moreover, TrustNet dynamically adjusts α_k after each epoch based on the observed learning performance of $\mathcal{F}(\boldsymbol{x}_k, \boldsymbol{\theta})$.

In detail we use cross-entropy H as standard loss measure to train our deep neural network $\mathcal{F}(\boldsymbol{x}_k, \boldsymbol{\theta})$:

$$\mathcal{H}(\mathcal{F}(\boldsymbol{x}_k, \boldsymbol{\theta}), y_k) = -\sum_{i=0}^{c-1} \mathbb{1}(y_k, c) \log \mathcal{F}(\boldsymbol{x}_k, \boldsymbol{\theta}) \tag{6}$$

where $\mathbb{1}(y_k, c)$ is an indicator function equal to 1 if $y_k = c$ and 0 otherwise. For each data point $\boldsymbol{x}_k$ in $\mathcal{D}'$, we assign weights of α_k and $(1 - \alpha_k)$ to the cross-entropy of the given y_k and inferred $\tilde{y}_k$ labels, respectively. We let $\alpha_k \in [0, 1]$. Hence, we write the robust loss function $\mathcal{L}_{robust}$ as following:

$$\begin{aligned} \mathcal{L}_{robust}(\mathcal{F}(\boldsymbol{x}_k, \boldsymbol{\theta}), y_k, \tilde{y}_k) = {} & \alpha_k \ \mathcal{H}(\mathcal{F}(\boldsymbol{x}_k, \boldsymbol{\theta}), y_k) \\ & + (1 - \alpha_k) \ \mathcal{H}(\mathcal{F}(\boldsymbol{x}_k, \boldsymbol{\theta}), \tilde{y}_k). \end{aligned} \tag{7}$$

When the weight factor is low, we put more weight on the cross-entropy of inferred labels, and vice versa. In the following, we explain how to dynamically set α_k per epoch.

4.3.1 Dynamic α_k. Here we adjust α_k based on the uncertainty of TrustNet and LABELNET. When the learning capacities of LABELNET and TrustNet are higher (lower values of loss function), we have more confidence on the inferred labels and put more weight on the second term of Eq. 7, i.e., smaller α_k values. As a rule of thumb, at the beginning α_k values are high since TrustNet experiences higher losses at the start of training. Then α_k values gradually decrease with the growing capacity of TrustNet.

Let $\alpha_{k,e}$ be the weight of the k^{th} image at epoch e. We initialize $\alpha_{k,0}$ based on the entropy value S from inferred class probabilities

```
Algorithm 1: TrustNet training
   Input  : Trusted dataset $\mathcal{T}$, Untrusted dataset $\mathcal{D}$; Epochs
            $E_{LABELNET}$,$E_{TrustNet}$.
            Untrusted dataset $\mathcal{D}$ made of: Observed samples $\boldsymbol{x}$,
            Given labels $y$
            Trusted dataset $\mathcal{T}$ made of: Observed samples $\boldsymbol{x}$,
            Given labels $y$, True labels $\hat{y}$
   Output : Trained TrustNet $\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta})$
 1 Initialize $\mathcal{F}^{\mathcal{A}}$ and $\mathcal{F}^{\mathcal{E}}$ with random $\boldsymbol{\theta}^{\mathcal{A}}$ and $\boldsymbol{\theta}^{\mathcal{E}}$
 2 for $e$ = 0, 1, ..., $E_{LABELNET}$ on $\mathcal{T}$ do
 3     Train $\mathcal{F}^{\mathcal{E}}$ and $\mathcal{F}^{\mathcal{A}}$                #LABELNET training
 4 end
 5 $\mathcal{D}' = \mathcal{D}$ extended with $\tilde{y} = \mathcal{F}^{\mathcal{E}}(<\mathcal{F}^{\mathcal{A}}(\boldsymbol{x}, \boldsymbol{\theta}^{\mathcal{A}}), y>, \boldsymbol{\theta}^{\mathcal{E}})$
    #LABELNET inference
 6 Initialize $\mathcal{F}$ with random $\boldsymbol{\theta}$             #TrustNet training
 7 for $e$ = 0, 1, ..., $E_{TrustNet}$ on $\mathcal{D}'$ do
 8     if $e == 0$ then
 9         $\alpha_{k,0} = S(\tilde{\boldsymbol{y}}_k)$
10     else
11         $\alpha_{k,e} = \alpha_{k,e-1}(1 + \frac{S(\boldsymbol{y}_k^{\mathcal{F}}(e)) - S(\boldsymbol{y}_k^{\mathcal{F}}(e-1))}{S(\boldsymbol{y}_k^{\mathcal{F}}(e-1))})$
12     end
13     Train $\mathcal{F}(\boldsymbol{x}, \boldsymbol{\theta}_e)$ with
        $\alpha_{k,e}\ \mathcal{H}(\mathcal{F}(\boldsymbol{x}_k, \boldsymbol{\theta}_e), y_k) + (1 - \alpha_{k,e})\ \mathcal{H}(\mathcal{F}(\boldsymbol{x}_k, \boldsymbol{\theta}_e), \tilde{y}_k)$
        for each sample $k$
14 end
```

$\tilde{\boldsymbol{y}}_k$ of LABELNET:

$$S(\tilde{\boldsymbol{y}}_k) = -\sum_{i=0}^{c-1} \tilde{y}_k^i \ log\ \tilde{y}_k^i$$

where c is the number of classes and $\tilde{y}_k^i$ is the i^{th} class probability of $\tilde{\boldsymbol{y}}_k$. We use LABELNET since we do not have yet any predictions from TrustNet's own neural network.

For subsequent epochs, $e > 0$, we switch to TrustNet as source of entropy values. We gradually adjust $\alpha_{k,e}$ based on the relative difference between current and previous epoch values:

$$\alpha_{k,e} = \alpha_{k,e-1}(1 + \frac{S(\boldsymbol{y}_k^{\mathcal{F}}(e)) - S(\boldsymbol{y}_k^{\mathcal{F}}(e-1))}{S(\boldsymbol{y}_k^{\mathcal{F}}(e-1))}) \quad \forall e > 0, \tag{8}$$

where $\boldsymbol{y}_k^{\mathcal{F}}(e)$ are the class probabilities predicted by $\mathcal{F}(\cdot, \boldsymbol{\theta})$ for the k^{th} image at epoch e. When the entropy values decrease, we gain more confidence in TrustNet and the weights on the inferred labels (1-$(1 - \alpha)$) increase.

We summarize the training procedure of TrustNet in Algorithm 1. Training LABELNET consists of training two neural networks: Expert, $\mathcal{F}^{\mathcal{E}}(\cdot, \boldsymbol{\theta}^{\mathcal{E}})$, and Amateur, $\mathcal{F}^{\mathcal{A}}(\cdot, \boldsymbol{\theta}^{\mathcal{A}})$, using the trusted data $\mathcal{T}$ for $E_{LABELNET}$ epochs (line 1-4). Then we need to compute the inferred labels for all data points in $\mathcal{D}$ to produce $\mathcal{D}'$ (line 5). Finally, we train TrustNet for $E_{TrustNet}$ epochs (line 6-14). The initialization of α_k is via the entropy of the inferred labels (line 9) and then updated by the entropy of predicted labels (line 11). The robust loss function is computed accordingly (line 13).

5 EVALUATION

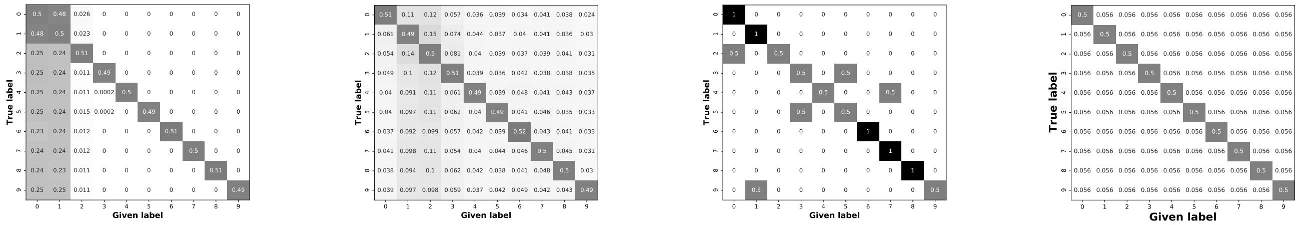
In this section, we empirically compare TrustNet against the state of the art noise, under both synthetic and real-world noises. We aim to show the effectiveness of TrustNet via testing accuracy on diverse and challenging noise patterns.

5.1 Experiments setup

We consider three datasets: CIFAR-10 [17], CIFAR-100 [17] and Clothing1M [38]. CIFAR-10 and CIFAR-100 both have 60K images of 32×32-pixels organized in 10 and 100 classes, respectively. These two datasets have no or minimal label noise. We split the datasets into 50K training and 10K testing sets and inject into the training set the label noises from Understanding DNNs. section. We assume that 10% of the training set forms the trusted data with access to the clean labels used as ground truth. We use this trusted set to learn the noise transition via LABELNET. In turn, LABELNET infers the estimated labels for the remaining training data. The whole training set is then used to train TrustNet. Clothing1M contains 1 million images scrapped from the Internet which we resize and crop to 224 × 224 pixels. Images are classified into 14 class labels. These labels are affected by real-world noise stemming from the automatic labelling. Out of the 1 million images, a subset of trusted expert-validated images contains the ground truth labels. This subset consists of 47K and 10K images for training and testing, respectively. As for CIFAR-10 and CIFAR-100, we use the trusted set to train LABELNET and infer the estimated labels for the rest of the dataset to train TrustNet. Note that for all three datasets, only training set is subject to label noise, not testing set.

The architecture of Expert consists of a 4-layer feed-forward neural network with Leaky ReLU activation functions in the hidden layers and sigmoid in the last layer. This Expert architecture is used across all datasets. TrustNet and Amateur use the same architecture, which depends on the dataset. For CIFAR-10 TrustNet and Amateur consist in an 8-layer CNN with 6 convolutional layers followed by 2 fully connected layers with ReLU activation functions as in [35]. For CIFAR-100 both rely on the ResNet44 architecture. Finally, Clothing1M uses pretrained ResNet101 with ImageNet. TrustNet (LABELNET) is trained for 120 (150) and 200 (180) for CIFAR-10 and CIFAR-100, respectively, using SGD optimizer with batch size 128, momentum 0.9, weight decay 10^{-4}, and learning rate 0.01. Finally, Clothing1M uses 50 (35) epochs and batch size 32, momentum 0.9, weight decay 5×10^{-3} and learning rate 2×10^{-3} divided by 10 every 5 epochs.

Our target evaluation metric is the accuracy achieved on the clean testing set, i.e. not affected by noise. We compare TrustNet against six noise resilient networks from the state of the art: SCL [36], D2L [37], Forward [25], Bootstrap [27], Co-teaching+ [41], and Co-teaching [10]. We do not compare TrustNet to LABELNET because primarily LABELNET requires labels in the inference process and training. Moreover, LABELNET needs accessing the ground truth for the whole data samples. All training uses Keras v2.2.4 and Tensorflow v1.13. We use 10% of the dataset as the trusted samples for the pre-training of baselines to have a fair comparison.



(a) Truncated normal, $\mu = 1, \sigma = 0.5$

(b) Bim., $\mu_1 = 2$ $\sigma_1 = 1$ $\mu_2 = 7$ $\sigma_2 = 3$

(c) Partial targeted

(d) Symmetric

Figure 3: Three study cases on CIFAR-10 with 10 classes that show the transition matrices for noise ratio $\varepsilon = 0.5$.

5.2 Synthetic Noise Patterns

For CIFAR-10 and CIFAR-100, we inject synthetic noise. We focus on asymmetric noise patterns following a truncated normal and bimodal distribution, and symmetric noise, as discussed in Understanding DNNs. section. We inject noises with average rates $\varepsilon = 0.4$, 0.5 and 0.6. For truncated normal the target classes and variances are class 1 with $\sigma = 0.5$ or $\sigma = 5$ and 10 with $\sigma = 1$ or $\sigma = 10$ for CIFAR-10 and CIFAR-100, respectively. For bimodal we use $\mu_1 = 2$, $\sigma_1 = 1$ plus $\mu_2 = 7$, $\sigma_2 = 3$ and $\mu_1 = 20$, $\sigma_1 = 10$ plus $\mu_2 = 70$, $\sigma_2 = 5$ for CIFAR-10 and CIFAR-100, respectively. We illustrate the noise transition matrix of these noise patterns in Figure 3.

5.2.1 *CIFAR-10.*

We summarize the results of CIFAR-10 in Table 1. We report the average and standard deviation across three runs. Overall the results are stable across different runs as seen from the low values of standard deviation. For readability reasons, we skip the results for 50% noise in the table. These results follow the trend between 40% and 60% noise. An extended table is shown in the appendix.

TrustNet achieves the highest accuracy for bimodal noises, which is one of the most difficult noise patterns based on Understanding DNNs. section. Here the accuracy of TrustNet is consistently the best beating the second best method by increasing 2.4%, 21.1%, and 27.2% for 40%, 50%, and 60% noise ratios, respectively. At the same time, TrustNet is the second best method for symmetric and truncated normal asymmetric noise. Here the best method is often SCL, which also leverages a modified loss function to enhance the per class accuracy using symmetric cross-entropy. This design targets direct symmetric noise where SCL outperforms TrustNet. Considering the asymmetric truncated normal noise, the difference is smaller and decreasing with increasing noise ratio. At 60% noise SCL is only marginally better by, on average, 2.9%. Finally, test accuracy variations are not noticeable with increasing σ values. All other baselines perform worse.

5.2.2 *CIFAR-100.*

Table 2 summarizes the CIFAR-100 results over three runs. CIFAR-100 is more challenging than CIFAR-10 because it increases tenfold the number of classes while keeping the same amount of training data. This is clearly reflected in the accuracy results across all methods, but TrustNet overall seems to be more resilient. Here, TrustNet achieves the highest accuracy for both asymmetric noise patterns under all considered noise ratios. On average, the accuracy of TrustNet is higher than SCL, the second best solution, by 2%. The improvement is higher for higher noise ratios and lower variation, i.e., $\sigma = 1$. SCL outperforms TrustNet on symmetric noise of low and middle intensity, i.e., $\varepsilon = [0.4, 0.5]$, but the difference diminishes with increasing noise, and at 60% TrustNet performs better. Different from CIFAR-10, test accuracy variations become noticeable for truncated normal noise with increasing σ values producing a positive effect across most baselines. All other baselines perform worse.

5.3 Real-world Noisy Data: Clothing1M

We use the noise pattern observed in real world data from the Clothing1M dataset to demonstrate the effectiveness and importance of estimating the noise transition matrix in TrustNet. Table 3 summarizes the results on the testing accuracy for TrustNet and the six baselines. The measured average noise ratio across all classes is 41%. Here, TrustNet achieves the highest accuracy, followed by SCL and Forward. Forward is another approach trying to estimate the noise transition matrix. The better accuracy of TrustNet is attributed to the additional label estimation from LABELNET learned via the trusted data and dynamically weighting the loss functions from given and inferred labels. The promising results here confirm that the novel learning algorithm of TrustNet can tackle challenging label noise patterns appearing in real-world datasets.

6 DISCUSSION

In this section, we discuss testing accuracy on clean and noisy samples. The analysis derived in Understanding DNNs. section consider testing on labels affected by the same noise as training data. This is due to the fact that the ground truth of labels is usually assumed unknown and not even available in the typical learning scenarios. However, the accuracy measured from the noisy testing data provides no information about how effective resilient networks defend the training process against the noisy data. Hence, related work on noisy label learning tests on clean samples, which show different trends as hinted in the evaluation section. Figure. 4 compares the two approaches across different noise patterns empirically. In general, in the case of clean test labels, the testing accuracy decreases with increasing noise ratios almost linearly. As for noisy labels, testing accuracy shows a clear quadratic trend, first decreasing before increasing again. Specifically, the lowest accuracy happens

Table 1: Accuracy on clean testing set for CIFAR-10 under 40% and 60% noise and patterns: i) symmetric, ii) bimodal with $\mu_1 = 2$, $\sigma_1 = 1$, $\mu_2 = 7$, $\sigma_2 = 3$ and iii) truncated normal with $\mu = 1$, $\sigma = [0.5, 5]$. Best results in bold.

CIFAR-10								
Methods	Symmetric		Bimodal Asymmetric		Truncated Normal Asymmetric			
	$\varepsilon = 0.4$	$\varepsilon = 0.6$	$\varepsilon = 0.4$	$\varepsilon = 0.6$	$\varepsilon = 0.4$		$\varepsilon = 0.6$	
					$\sigma = 0.5$	$\sigma = 5$	$\sigma = 0.5$	$\sigma = 5$
TrustNet	77.03 ± 0.32	61.22 ± 0.66	**72.67 ± 0.33**	**42.18 ± 0.61**	74.21 ± 0.69	73.88 ± 0.78	66.48 ± 0.61	67.23 ± 0.57
SCL	**81.50 ± 0.22**	**73.13 ± 0.12**	69.07 ± 1.17	15.00 ± 0.67	**80.93 ± 0.50**	**80.90 ± 0.14**	**68.67 ± 0.96**	**70.90 ± 0.67**
D2L	75.87 ± 0.33	60.54 ± 0.44	70.59. ± 0.11	34.67 ± 0.36	70.01 ± 0.21	71.22 ± 0.57	59.62 ± 0.13	62.35 ± 0.43
Forward	68.40 ± 0.36	51.27 ± 1.11	61.03 ± 0.61	33.27 ± 0.53	67.83 ± 0.86	68.63 ± 0.65	50.90 ± 0.99	51.53 ± 0.74
Bootstrap	71.03 ± 0.85	56.47 ± 1.18	61.10 ± 0.54	31.17 ± 0.59	70.80 ± 0.78	71.07 ± 0.78	54.87 ± 0.50	55.80 ± 1.23
Co-teaching+	72.44 ± 0.37	60.08 ± 0.48	55.33 ± 0.19	38.37 ± 0.77	57.02 ± 0.45	59.81 ± 0.72	41.11 ± 0.36	43.16 ± 0.29
Co-teaching	72.04 ± 0.61	58.78 ± 0.32	53.89 ± 0.25	37.51 ± 0.18	55.41 ± 0.19	58.31 ± 0.41	40.06 ± 0.69	41.95 ± 0.61

Table 2: Accuracy on clean testing set for CIFAR-100 under 40% and 60% noise and patterns: i) symmetric, ii) bimodal with $\mu_1 = 20$, $\sigma_1 = 10$, $\mu_2 = 70$, $\sigma_2 = 5$, and iii) truncated normal with $\mu = 10$, $\sigma = [1, 10]$. Best results in bold.

CIFAR-100								
Methods	Symmetric		Bimodal Asymmetric		Truncated Normal Asymmetric			
	$\varepsilon = 0.4$	$\varepsilon = 0.6$	$\varepsilon = 0.4$	$\varepsilon = 0.6$	$\varepsilon = 0.4$		$\varepsilon = 0.6$	
					$\sigma = 1$	$\sigma = 10$	$\sigma = 1$	$\sigma = 10$
TrustNet	41.23 ± 0.43	**29.11 ± 0.12**	**45.01 ± 0.14**	**32.32 ± 0.30**	**37.66 ± 0.36**	**44.56 ± 0.42**	**23.96 ± 0.38**	**33.29 ± 0.41**
SCL	**42.30 ± 0.36**	28.43 ± 0.69	43.57 ± 0.42	30.70 ± 0.88	37.63 ± 0.62	43.50 ± 0.45	19.20 ± 0.57	31.93 ± 0.39
D2L	41.01 ± 0.21	21.41 ± 0.12	32.47 ± 0.43	10.55 ± 0.19	10.66 ± 0.16	10.32 ± 0.21	10.11 ± 0.38	10.05 ± 0.14
Forward	36.40 ± 0.37	16.00 ± 0.80	38.80 ± 0.28	19.03 ± 0.69	34.03 ± 0.33	39.80 ± 0.33	10.27 ± 0.47	22.90 ± 0.00
Bootstrap	28.40 ± 0.16	6.70 ± 0.59	32.17 ± 0.62	10.10 ± 0.94	27.23 ± 0.71	34.17 ± 0.96	6.10 ± 0.16	12.53 ± 1.84
Co-teaching+	39.35 ± 0.35	26.32 ± 0.54	34.64 ± 0.59	26.52 ± 0.58	34.17 ± 0.24	36.59 ± 0.32	18.24 ± 0.71	26.61 ± 0.33
Co-teaching	37.82 ± 0.22	25.44 ± 0.71	33.76 ± 0.54	26.12 ± 0.33	32.02 ± 0.56	33.85 ± 0.62	16.99 ± 0.32	25.33 ± 0.12

Table 3: Accuracy on clean testing set of real-world noisy Clothing1M.

Methods	**TrustNet**	**SCL**	**D2L**	**Forward**	**Bootstrap**	**Co-teaching+**	**Co-teaching**
Accuracy(%)	**73.06**	70.78	69.43	70.04	68.77	70.33	70.10

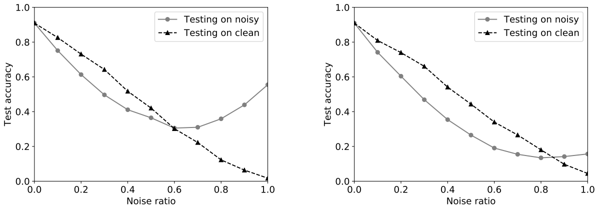
at noise ratio of 0.6 and 0.8 in the case of the truncated normal noise example with $\mu = 1$ and $\sigma = 0.5$ (Figure. 4a), and the bimodal noise example with $\mu_1 = 2, \sigma_1 = 0.5, \mu_2 = 7, \sigma_2 = 5$ (Figure. 4b), respectively. The reason is that specific class examples with erroneous labels become more numerous than examples with the true class, e.g., more truck images are labelled as an automobile than automobile images. Such an effect is missing when testing on clean labels.

7 CONCLUSION

Motivated by the disparity of label noise patterns studied in the prior state-of-the-art methods, we first derive the analytical understanding of synthetic and real-world noise, i.e., how testing accuracy degrades with noise ratios and patterns. Challenging noise patterns identified here lead to the proposed learning framework, TrustNet, which noise resilient classification. TrustNet first learns a noise transition matrix via a small set of trusted data and LABELNET. Combining the estimated labels inferred from LABELNET, TrustNet computes a robust loss function from both given and inferred labels via dynamic weights according to the learning confidence, i.e., the entropy. The proposed method estimates the correct labels of various datasets with different sizes ranging from small to large. We evaluate TrustNet on CIFAR-10, CIFAR-100, and Clothing1M using a diverse set of synthetic and real-world noise patterns. The higher testing accuracy against state-of-the-art resilient networks shows that TrustNet can effectively learn the noise transition and enhance the robustness of loss function against noisy labels.

ACKNOWLEDGMENTS

This work has been partly funded by the Swiss National Science Foundation NRP75 project 407540_167266.



(a) Truncated normal (b) Bimodal

Figure 4: Empirical testing on noisy and clean labeled data on CIFAR-10.

REFERENCES

[1] Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. 2017. A closer look at memorization in deep networks. In *ICML*. 233–242.

[2] Pengfei Chen, Benben Liao, Guangyong Chen, and Shengyu Zhang. 2019. Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels. In *ICML*. 1062–1070.

[3] Filipe R. Cordeiro and Gustavo Carneiro. 2020. A Survey on Deep Learning with Noisy Labels: How to train your model when you cannot trust on the annotations?. In *SIBGRAPI*. IEEE, 9–16.

[4] Benoît Frénay and Michel Verleysen. 2013. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems* 25, 5 (2013), 845–869.

[5] Amirmasoud Ghiassi, Robert Birke, Rui Han, and Lydia Y.Chen. 2021. LABELNET: Recovering Noisy Labels. In *IJCNN*.

[6] Amirmasoud Ghiassi, Taraneh Younesian, Zhilong Zhao, Robert Birke, Valerio Schiavoni, and Lydia Y Chen. 2019. Robust (deep) learning framework against dirty labels and beyond. In *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 236–244.

[7] Jacob Goldberger and Ehud Ben-Reuven. 2017. Training deep neural-networks using a noise adaptation layer. In *ICLR*.

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR*.

[9] Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor W. Tsang, Ya Zhang, and Masashi Sugiyama. 2018. Masking: A New Perspective of Noisy Supervision. In *NIPS*. 5841–5851.

[10] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NIPS*. 8527–8537.

[11] Jiangfan Han, Ping Luo, and Xiaogang Wang. 2019. Deep Self-Learning From Noisy Labels. In *ICCV*. 5137–5146.

[12] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. 2018. Using trusted data to train deep networks on labels corrupted by severe noise. In *NIPS*. 10456–10465.

[13] Sara Hooker, Yann Dauphin, Aaron Courville, and Andrea Frome. 2019. Selective brain damage: Measuring the disparate impact of model pruning. (2019).

[14] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. MentorNet: Learning Data-Driven Curriculum for Very Deep Neural Networks on Corrupted Labels. In *ICML*. 2309–2318.

[15] Takuhiro Kaneko, Yoshitaka Ushiku, and Tatsuya Harada. 2019. Label-Noise Robust Generative Adversarial Networks. In *CVPR*.

[16] Nikola Konstantinov and Christoph Lampert. 2019. Robust Learning from Untrusted Sources. In *ICML*, Vol. 97. 3488–3498.

[17] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. 2009. CIFAR-10/100 (Canadian Institute for Advanced Research). (2009). http://www.cs.toronto.edu/~kriz/cifar.html

[18] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. 2018. CleanNet: Transfer Learning for Scalable Image Classifier Training With Label Noise. In *CVPR*. 5447–5456.

[19] Wen Li, Limin Wang, Wei Li, Eirikur Agustsson, and Luc Van Gool. 2017. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862* (2017).

[20] Guanxiong Liu, Issa Khalil, and Abdallah Khreishah. 2019. ZK-GanDef: A GAN Based Zero Knowledge Adversarial Training Defense for Neural Networks. In *IEEE/IFIP DSN*. 64–75.

[21] Xingjun Ma, Yisen Wang, Michael E. Houle, Shuo Zhou, Sarah M. Erfani, Shu-Tao Xia, Sudanthi N. R. Wijewickrema, and James Bailey. 2018. Dimensionality-Driven Learning with Noisy Labels. In *ICML*. 3361–3370.

[22] Eran Malach and Shai Shalev-Shwartz. 2017. Decoupling" when to update" from" how to update". In *NIPS*. 960–970.

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

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

[25] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *IEEE CVPR*. 1944–1952.

[26] Jennifer Prendki. 2018. The curse of big data labeling and three ways to solve it. https://aws.amazon.com/blogs/apn/the-curse-of-big-data-labeling-and-three-ways-to-solve-it/ Accessed on 22.08.2021.

[27] Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training Deep Neural Networks on Noisy Labels with Bootstrapping. In *ICLR Workshop*.

[28] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. 2018. Learning to Reweight Examples for Robust Deep Learning. *ICML* (2018).

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision* 115, 3 (2015), 211–252.

[30] Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. 2020. Evaluating machine accuracy on imagenet. In *ICMLg*. PMLR, 8634–8644.

[31] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. 2019. Meta-weight-net: Learning an explicit mapping for sample weighting. In *NIPS*. 1919–1930.

[32] Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. 2014. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080* (2014).

[33] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. 2020. From imagenet to image classification: Contextualizing progress on benchmarks. In *ICML*. PMLR, 9625–9635.

[34] Arash Vahdat. 2017. Toward Robustness against Label Noise in Training Deep Discriminative Neural Networks. In *NIPS*. 5596–5605.

[35] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. 2018. Iterative learning with open-set noisy labels. In *IEEE CVPR*. 8688–8696.

[36] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. 2019. Symmetric cross entropy for robust learning with noisy labels. In *IEEE ICCV*. 322–330.

[37] Yisen Wang, Xingjun Ma, Michael E Houle, Shu-Tao Xia, and James Bailey. 2018. Dimensionality-driven learning with noisy labels. *ICML* (2018), 3361–3370.

[38] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. 2015. Learning from massive noisy labeled data for image classification. In *IEEE CVPR*. 2691–2699.

[39] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. 2020. Dual t: Reducing estimation error for transition matrix in label-noise learning. *NIPS* 33 (2020).

[40] Kun Yi and Jianxin Wu. 2019. Probabilistic end-to-end noise correction for learning with noisy labels. In *CVPR*. 7017–7025.

[41] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W Tsang, and Masashi Sugiyama. 2019. How does disagreement help generalization against label corruption?. In *ICML*. 7164–7173.

[42] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *ICLR*.

[43] Zhilu Zhang and Mert R. Sabuncu. 2018. Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels. In *NIPS*. 8792–8802.

A TEST ACCURACY OF CIFAR-10 AND CIFAR-100 UNDER 50% NOISE RATIO

We extend the results in Table 1 and Table 2 by adding the test accuracy TrustNet and other rivals for different noise patterns with 50% noise ratio. The results are shown in Table 4 and Table 5 for CIFAR-10 and CIFAR-100, respectively. In Table 4, TrustNet performs better comparing to other competitors under $\varepsilon = 0.5$ for bimodal noise. As we mentioned in § 5, our method outperforms against rivals for all the cases except symmetric noise with $\varepsilon = 0.4$ and $\varepsilon = 0.5$ for CIFAR-100.

B PROOF OF LEMMA 1

Lemma 1. *For noise with fixed noise ratio ε and any given label distribution with probability function $P(y = j), \forall j \neq i$, where $i \in C$ is the true label, the test accuracy is*

$$P(y^{\mathcal{F}} = y) = (1 - \varepsilon)^2 + \varepsilon^2 \sum_{j \neq i}^{C} P^2(y = j) \tag{9}$$

Proof. We have that $T_{ii} = 1 - \varepsilon, \forall i \in C$ since all classes are affected by the same noise ratio. Moreover, the probability of selecting noisy class labels is scaled by the noise ratio $T_{ij} = \varepsilon \ P(y = j), j \neq i \in C$. Now:

$$\begin{aligned} P(y^{\mathcal{F}} = y) &= \sum_{i}^{C} P(\hat{y} = i) P(y^{\mathcal{F}} = y | \hat{y} = i) \\ &= \sum_{i}^{C} P(\hat{y} = i) \sum_{j}^{C} T_{ij}^2 \\ &= \sum_{i}^{C} P(\hat{y} = i) [T_{ii}^2 + \sum_{j \neq i}^{C} T_{ij}^2] \\ &= \sum_{i}^{C} P(\hat{y} = i) [(1 - \varepsilon)^2 + \varepsilon^2 \sum_{j \neq i}^{C} P^2(y = j)]. \end{aligned} \tag{10}$$

Since $\sum_{i}^{C} P(\hat{y} = i) = 1$, we obtain Eq. 1. ■

C PROOF OF LEMMA 2

Lemma 2. *For partial class noise with equal class label probability, where S is the set affected by noise with ratio ε and U is the set of unaffected labels, for any true label $i \in C$ and any given label distribution with probability function $P(y = j), \forall j \neq i$, the test accuracy is*

$$P(y^{\mathcal{F}} = y) = \frac{|U|}{|C|} + \frac{|S|}{|C|} [(1 - \varepsilon)^2 + \varepsilon^2 \sum_{j \neq i}^{S} P^2(y = j)] \tag{11}$$

Proof. We have that for affected labels in S the same noise transition definitions hold, i.e. $T_{ii} = 1 - \varepsilon, \forall i \in S$ and $T_{ij} = \varepsilon \ P(y = j), j \neq i \in S$. For unaffected labels we have that $\varepsilon = 0$ hence $T_{ii} = 1, \forall i \in U$ and $T_{ij} = 0, j \neq i \in U$. Moreover, $P(\hat{y} = i) = \frac{1}{|C|}$ assuming all class labels are equally probable. Now:

$$\begin{aligned} P(y^f = y) &= \sum_{i}^{C} P(\hat{y} = i) P(y^f = y | \hat{y} = i) \\ &= \sum_{i}^{|U|} P(\hat{y} = i) P(y^f = y | \hat{y} = i) \\ &+ \sum_{i'}^{|S|} P(\hat{y} = i') P(y^f = y | \hat{y} = i') \\ &= \sum_{i}^{U} P(\hat{y} = i) \sum_{j}^{U} T_{ij}^2 + \sum_{i'}^{S} P(\hat{y} = i') \sum_{j'}^{S} T_{i'j'}^2 \\ &= \sum_{i}^{U} P(\hat{y} = i) [T_{ii}^2 + \sum_{j \neq i}^{U} T_{ij}^2] \\ &+ \sum_{i'}^{S} P(\hat{y} = i') [T_{i'i'}^2 + \sum_{j' \neq i'}^{S} T_{i'j'}^2] \\ &= \frac{1}{|C|} \sum_{i}^{U} [T_{ii}^2 + \sum_{j \neq i}^{U} T_{ij}^2] \\ &+ \frac{1}{|C|} \sum_{i'}^{S} [T_{i'i'}^2 + \sum_{j' \neq i'}^{S} T_{i'j'}^2] \\ &= \frac{1}{|C|} \sum_{i}^{U} 1 + \frac{1}{|C|} \sum_{i'}^{S} [(1 - \varepsilon)^2 + \varepsilon^2 \sum_{j' \neq i'}^{S} P^2(y = j')] \\ &= \frac{|U|}{|C|} + \frac{|S|}{|C|} [(1 - \varepsilon)^2 + \varepsilon^2 \sum_{j' \neq i'}^{S} P^2(y = j')] \end{aligned}$$

■

Table 4: Accuracy on clean testing set for CIFAR-10 under 40%, 50%, and 60% noise and patterns: i) symmetric, ii) bimodal with $\mu_1 = 2$, $\sigma_1 = 1$, $\mu_2 = 7$, $\sigma_2 = 3$, and iii) truncated normal with $\mu = 1$, $\sigma = [0.5, 5]$. Best results in bold.

Methods	Symmetric			Bimodal Asymmetric			Normal Asymmetric					
	$\varepsilon = 0.4$	$\varepsilon = 0.5$	$\varepsilon = 0.6$	$\varepsilon = 0.4$	$\varepsilon = 0.5$	$\varepsilon = 0.6$	$\varepsilon = 0.4$		$\varepsilon = 0.5$		$\varepsilon = 0.6$	
							$\sigma = 0.5$	$\sigma = 5$	$\sigma = 0.5$	$\sigma = 5$	$\sigma = 0.5$	$\sigma = 5$
TrustNet	77.03 ± 0.32	69.87 ± 0.56	61.22 ± 0.66	**72.67 ± 0.33**	**67.28 ± 0.57**	**42.18 ± 0.61**	74.21 ± 0.69	73.88 ± 0.78	69.34 ± 0.44	69.52 ± 0.75	66.48 ± 0.61	67.23 ± 0.57
SCL	**81.50 ± 0.22**	**78.17 ± 0.76**	**73.13 ± 0.12**	69.07 ± 1.17	46.13 ± 1.10	15.00 ± 0.67	**80.93 ± 0.50**	**80.90 ± 0.14**	**76.80 ± 0.45**	**77.33 ± 0.45**	**68.67 ± 0.96**	**70.90 ± 0.67**
D2L	75.87 ± 0.33	66.34 ± 1.43	60.54 ± 0.44	70.59. ± 0.11	52.84 ± 0.43	34.67 ± 0.36	70.01 ± 0.21	71.22 ± 0.57	65.32 ± 0.07	66.08 ± 0.38	59.62 ± 0.13	62.35 ± 0.43
Forward	68.40 ± 0.36	61.77 ± 0.21	51.27 ± 1.11	61.03 ± 0.61	46.37 ± 0.33	33.27 ± 0.53	67.83 ± 0.86	68.63 ± 0.65	61.20 ± 0.22	61.40 ± 0.16	50.90 ± 0.99	51.53 ± 0.74
Bootstrap	71.03 ± 0.85	65.33 ± 0.41	56.47 ± 1.18	61.10 ± 0.54	45.97 ± 0.34	31.17 ± 0.59	70.80 ± 0.78	71.07 ± 0.78	64.13 ± 0.60	65.43 ± 0.09	54.87 ± 0.50	55.80 ± 1.23
Co-teaching+	72.44 ± 0.37	65.82 ± 0.23	60.08 ± 0.48	55.33 ± 0.19	42.06 ± 0.53	38.37 ± 0.77	57.02 ± 0.45	59.81 ± 0.72	51.56 ± 0.62	52.73 ± 0.38	41.11 ± 0.36	43.16 ± 0.29
Co-teaching	72.04 ± 0.61	63.78 ± 0.13	58.78 ± 0.32	53.89 ± 0.25	40.67 ± 0.72	37.51 ± 0.18	55.41 ± 0.19	58.31 ± 0.41	50.29 ± 0.61	50.83 ± 0.41	40.06 ± 0.69	41.95 ± 0.61

Table 5: Accuracy on clean testing set for CIFAR-100 under 40%, 50%, and 60% noise and patterns: i) symmetric, ii) bimodal with $\mu_1 = 20$, $\sigma_1 = 10$, $\mu_2 = 70$, $\sigma_2 = 5$, and iii) truncated normal with $\mu = 10$, $\sigma = [1, 10]$. Best results in bold.

Methods	Symmetric			Bimodal Asymmetric			Normal Asymmetric					
	$\varepsilon = 0.4$	$\varepsilon = 0.5$	$\varepsilon = 0.6$	$\varepsilon = 0.4$	$\varepsilon = 0.5$	$\varepsilon = 0.6$	$\varepsilon = 0.4$		$\varepsilon = 0.5$		$\varepsilon = 0.6$	
							$\sigma = 1$	$\sigma = 10$	$\sigma = 1$	$\sigma = 10$	$\sigma = 1$	$\sigma = 10$
TrustNet	41.23 ± 0.43	35.77 ± 0.28	**29.11 ± 0.12**	**45.01 ± 0.14**	**39.55 ± 0.62**	**32.32 ± 0.30**	**37.66 ± 0.36**	**44.56 ± 0.42**	**32.76 ± 0.13**	**39.04 ± 0.55**	**23.96 ± 0.38**	**33.29 ± 0.41**
SCL	**42.30 ± 0.36**	**35.93 ± 0.17**	28.43 ± 0.69	43.57 ± 0.42	37.60 ± 0.45	30.70 ± 0.88	37.63 ± 0.62	43.50 ± 0.45	29.77 ± 0.33	37.67 ± 0.74	19.20 ± 0.57	31.93 ± 0.39
D2L	41.01 ± 0.21	33.72 ± 0.34	21.41 ± 0.12	32.47 ± 0.43	21.23 ± 0.13	10.55 ± 0.19	10.66 ± 0.16	10.32 ± 0.21	10.14 ± 0.10	10.02 ± 0.26	10.11 ± 0.38	10.05 ± 0.14
Forward	36.40 ± 0.37	26.03 ± 0.97	16.00 ± 0.80	38.80 ± 0.28	30.13 ± 0.25	19.03 ± 0.69	34.03 ± 0.33	39.80 ± 0.33	21.23 ± 0.34	32.23 ± 0.31	10.27 ± 0.47	22.90 ± 0.00
Bootstrap	28.40 ± 0.16	14.37 ± 0.12	6.70 ± 0.59	32.17 ± 0.62	19.17 ± 0.86	10.10 ± 0.94	27.23 ± 0.71	34.17 ± 0.96	13.80 ± 0.41	22.97 ± 1.54	6.10 ± 0.16	12.53 ± 1.84
Co-teaching+	39.35 ± 0.35	33.77 ± 0.49	26.32 ± 0.54	34.64 ± 0.59	30.34 ± 0.24	26.52 ± 0.58	34.17 ± 0.24	36.59 ± 0.32	29.06 ± 0.52	33.30 ± 0.39	18.24 ± 0.71	26.61 ± 0.33
Co-teaching	37.82 ± 0.22	31.69 ± 0.61	25.44 ± 0.71	33.76 ± 0.54	28.89 ± 0.18	26.12 ± 0.33	32.02 ± 0.56	33.85 ± 0.62	28.01 ± 0.42	31.57 ± 0.26	16.99 ± 0.32	25.33 ± 0.12