Robust Learning via Golden Symmetric Loss of (un)Trusted Labels

Ghiassi, Amirmasoud; Birke, Robert; Chen, Lydia Y.

Publication date
2023

Document Version
Final published version

Published in
2023 SIAM International Conference on Data Mining, SDM 2023

Robust Learning via Golden Symmetric Loss of (un)Trusted Labels

Amirmasoud Ghiassi* s.ghiassi@tudelft.nl
Robert Birke† robert.birke@unito.it
Lydia Y. Chen* lydiaychen@ieee.com

Abstract

Learning robust deep models against noisy labels becomes ever critical when today's data is commonly collected from open platforms and subject to adversarial corruption. The information on the label corruption process, i.e., corruption matrix, can greatly enhance the robustness of deep models but still fall behind in combating hard classes. In this paper, we propose to construct a golden symmetric loss (GSL) based on the estimated corruption matrix as to avoid overfitting to noisy labels and learn effectively from hard classes. GSL is the weighted sum of the corrected regular cross entropy and reverse cross entropy. By leveraging a small fraction of trusted clean data, we estimate the corruption matrix and use it to correct the loss as well as to determine the weights of GSL. We theoretically prove the robustness of the proposed loss function in the presence of dirty labels. We provide a heuristics to adaptively tune the loss weights of GSL according to the noise rate and diversity measured from the dataset. We evaluate our proposed golden symmetric loss on both vision and natural language deep models subject to different types of label noise patterns. Empirical results show that GSL can significantly outperform the existing robust training methods on different noise patterns, showing accuracy improvement up to 18% on CIFAR-100 and 1% on real world noisy dataset of Clothing1M.

Keywords: Robust training, Deep learning models, Symmetric loss function, Noisy labels.

1 Introduction

Diverse datasets collected from the public domain which power up deep learning models present new challenges – highly noisy labels. It is not only time consuming to collect labels but also difficult to ensure a consistent label quality due to various annotation errors [1] and adversarial attacks [2]. The large capacity of deep learning models enables effective learning from complex datasets but also suffers from overfitting to the noise structure in the dataset. The curse of memorization effect [3] can degrade the accuracy of deep learning models in the presence of highly noisy labels. For example, in [4] the accuracy of AlexNet to classify CIFAR-10 images drops from 77% to 10%, when there are randomly flipped labels.

Designing learning models that can robustly train on noisy labels is thus imperative. To distill the impact of noisy labels, the related work either filters out suspiciously noisy data, derives robust loss functions or tries to proactively correct labels. Symmetric Cross entropy Loss (SCL) is shown effective in combating label noise especially for hard classes by combing the regular with the reverse cross entropy. The former avoids overfitting and the latter is resilient to label noise. Given its promising results, there is yet to have a clear principle on how to weight the regular and reverse cross entropy terms, e.g., at different noise rates and patterns. In contrast, Distilling [5] and Golden Loss Correction (GLC) [6] advocate to use a small clean data to improve the estimated corruption matrix. Specifically, GLC trains the deep model on both a clean and noisy set, whose loss is corrected through the corruption matrix. While the clean set is evenly chosen from all classes, the corrupted labels may appear unevenly across classes depending on the noise pattern [7, 8]. As the corrected loss of GLC does not differentiate the difficulty of classes, it may not learn those hard classes effectively.

We propose GSL to construct the golden symmetric loss that dynamically weights regular/reverse cross entropy and corrects the label prediction based on the estimated corruption matrix. Similar to GLC, GSL leverages clean data to estimate the corruption matrix which is used to correct labels and decide the weights of the golden symmetric loss. As such, GSL can effectively differentiate the difficulty level of classes by adjusting the weights and mitigate the impact of noise overfitting via the golden symmetric cross entropy. Specifically, we use the noise rate and noise diversity to adaptively tune the weights of modified cross entropy and reverse cross entropy. We prove that modified cross entropy by using corruption matrix is noise tolerant same as the reverse cross entropy. Empirical evaluation on vision and text datasets shows that GSL outperforms

*EEMCS, Delft University of Technology, The Netherlands
†DI - University of Torino, CINI HPC-KTT lab, Turin, Italy

the state-of-the-art methods under tested noise ratios from 0% to 100% for text datasets and noise ratios 30% and 60% for vision datasets. In addition, we illustrate that combining symmetric loss function and the corruption matrix estimation with correct dynamic weighting function is the best combination of robust methods against noisy label data.

The contributions of this paper are summarized as follows:

- We design a noise resilient method that estimates the corruption matrix using a small proportion of trusted data and then corrects the wrong prediction into the symmetric cross-entropy loss function.
- Using noise properties, including rate and diversity, we design a weighting function for the symmetric loss function to adjust the weights of improved cross-entropy and reverse cross-entropy adaptively.
- We compare GSL against state-of-the-art methods under noisy labels on the real-world dataset and synthetic vision and text datasets.

1.1 Motivation example We demonstrate the advantages and disadvantages of GLC and SCL, and the their combination (the proposed GSL) through the example of learning convolution networks on CIFAR-10 injected with 60% symmetric noise. The experimental setup is detailed in §6. Figure 1 shows the corruption matrix of the injected noise and the confusion matrices from the predictions of SCL, GLC, and GSL. Even if the injected noise is symmetric across all classes (see Figure 1(a)), prediction errors are distributed asymmetrically across the classes (see Figure 1(b), Figure 1(c) and Figure 1(d)). Though GLC can achieve a lower average error rate than SCL (reflected in darker diagonal elements on average), it performs worse in hard classes, e.g., class 4 (cat) and class 6 (dog) (difference in blue shades across the diagonal elements). By setting up proper weights for two types of cross entropy, GSL is able to achieve both superior average and per class accuracy.

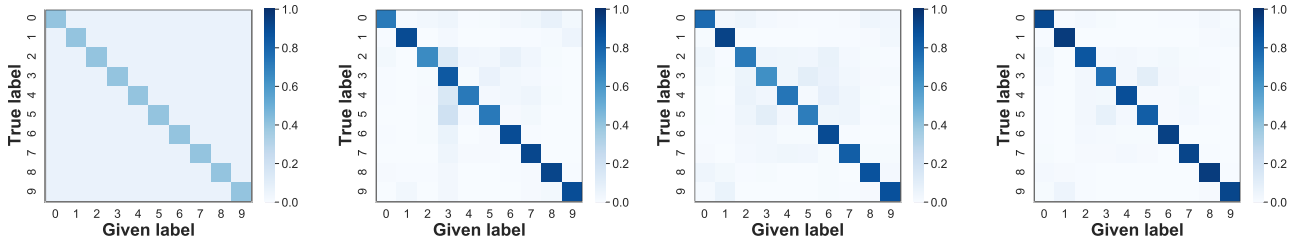
2 Related Work

Enhancing the robustness of deep models against noisy labels is an active research area. The massive datasets needed to train deep models are commonly found corrupted, [9], severely degrading the achievable accuracy, [4]. The impact of label noise on deep neural networks is first characterized by the theoretical testing accuracy over a limited set of noise patterns [10]. [11] suggest an undirected graph model for modeling label noise in deep neural networks and indicate symmetric noise to be more challenging than asymmetric. Current solutions can be categorized into three directions: (i) filtering out noisy labels: [12, 13]; (ii) correcting noisy labels: [1, 6, 5, 14, 15]; and (iii) deriving noise resilient loss functions: [16, 17, 18].

2.1 Noise Resilient Loss Function The loss function is modified to enhance the robustness to label noise by introducing new loss functions, [19, 20], or adjusting the weights of noisy data instances, [21, 18, 22]. Mean Absolute Error (MAE) [19, 23] and General Cross Entropy loss [23] are proposed as a noise resilient alternative but at the cost of slow convergence. To avoid overfitting to noise, D2L [22] uses the subspace dimensionality to assign weights to each data point, whereas Konstantinov [18] determines the loss weights based on the trustworthiness level of data sources. [20] propose symmetric cross-entropy loss that combines a new term of reverse cross entropy with traditional cross entropy via constant weights on both terms. Meta-Weight-Net [24] re-weights samples during optimizing loss function in the training process by using a multi-layer perceptron to predict the weight of each sample. With the same perspective, [25] uses the similarity of samples to the clean instances in the validation set for re-weighting them in loss function.

2.2 Label correction To avoid the data reduction caused by filtering, label correction methods adjust the predicted/given labels by using only noisy labels [1, 26, 27] or jointly with a small fraction of trusted data [28, 29, 5, 6]. [30] train the classifier by the "new" labels combining the raw and predicted labels without access to label ground truth. [1] estimate the noise corruption matrix by first training a classifier on the noisy labels and then using the softmax probabilities. [28] acquire human-verified labels to train a cleaning network for correcting noisy labels of multi-label classification problems. [29] estimate the noise transition probability by incorporating human assistance. [5] and [6] leverage a small set of clean data to estimate noise corruption matrix from the clean and noisy sets, respectively. DivideMix [31] is a semi-supervised method, including two networks and Gaussian Mixture Model for sample selection.

The proposed GSL combines resilient loss function and label correction by curating a small fraction of trusted data. We solicit a subset of informative data instances to estimate the corruption matrix and provide a minimum supervision on noisy labels. We also provide a heuristic to adaptively tune the weights of golden symmetric loss according to the noise characteristics of the dataset.



(a) Corruption matrix (b) GLC (c) SCL (d) GSL

Figure 1: Noise corruption matrix and confusion matrices of predictions for CIFAR-10 with 60% symmetric label noise.

3 Golden Symmetric Loss

Consider the classification problem having dataset $\tilde{\mathcal{D}} = \{(\boldsymbol{x}_n, \tilde{y}_n)\}_{n=1}^N$ where $\boldsymbol{x}_n \in \mathcal{X} \subset \mathbb{R}^d$ denotes the n^{th} observed sample, and $\tilde{y}_n \in \mathcal{Y} := \{1, ..., K\}$ the corresponding given label over K classes. Hereon n is ignored for the simplicity. $\tilde{y}$ is affected by label noise. The label corruption process is characterised by a corruption matrix $C_{ij} = P(\tilde{y} = j|y = i)$ for $i = 1, \ldots, K$ and $j = 1, \ldots, K$ where y is the true label. Synthetic noise patterns are expressed as a label corruption probability ε plus a noise label distribution. Let $g(\cdot, \boldsymbol{\theta})$ denote a neural network-based classifier parameterized by $\boldsymbol{\theta}$. For each data point $\boldsymbol{x}$, $f(\cdot, \boldsymbol{\theta})$ predicts the probability for each class label k: $p(k|\boldsymbol{x}) = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}}$ where z_j are the logits.

3.1 Symmetric Cross Entropy Let $q(k|\boldsymbol{x})$ denote the ground truth probability distribution over the K class labels where $q(k|\boldsymbol{x}) = 1$ for k equal to the true class y and $q(k|\boldsymbol{x}) = 0$ for all $k \neq y$. The cross entropy loss (ℓ_{ce}) and reverse cross entropy loss[1] (ℓ_{rce}) for $\boldsymbol{x}$ are:

$$\ell_{ce} = -\sum_{k=1}^{K} q(k|\boldsymbol{x}) \log p(k|\boldsymbol{x}), \tag{3.1}$$

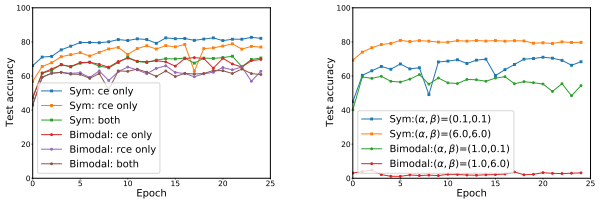
$$\ell_{rce} = -\sum_{k=1}^{K} p(k|\boldsymbol{x}) \log q(k|\boldsymbol{x}). \tag{3.2}$$

[20] combine cross entropy and reverse cross entropy into the symmetric cross entropy:

$$l_{sl} = \alpha\, \ell_{ce} + \beta\, \ell_{rce}. \tag{3.3}$$

where α and β are hyperparameters. On the one hand cross entropy loss is not robust to noise [19] but achieves good convergence [23]. On the other hand reverse cross entropy is tolerant to noise [20].

[1]To avoid problems with the logarithm, zero values of q are replaced by a small positive value, i.e. 10^{-4}.



(a) Different loss corrections ($\varepsilon = 70\%$) (b) α, β-tuning: best/worst ($\varepsilon = 80\%$)

Figure 2: Impact of loss correction and α, β-tuning on a 2-layer FC network trained on Twitter data.

3.2 Estimating Noise Corruption Matrix We estimate the noise corruption matrix as in [6]. The method fosters training a first classifier $g(\cdot, \theta)$ on noisy data to approximate the elements $\boldsymbol{C}_{ij}$ of the noise corruption matrix via a small fraction of trusted data $\mathcal{D}$ with known true label y. Practically given A_i the subset of trusted data with label of class i $\{A_i \subset \mathcal{D} : y = i\}$, the elements of $\boldsymbol{C}$ can be approximated by:

$$\hat{\boldsymbol{C}}_{ij} = P(\tilde{y} = j|y = i) \approx \frac{1}{|A_i|} \sum_{x \in A_i} g(\tilde{y} = j|\boldsymbol{x}, \Theta) \tag{3.4}$$

where $g(\tilde{y} = j|\boldsymbol{x}, \Theta)$ denotes predicted probability of $\boldsymbol{x}$ having class label j. That is $\hat{\boldsymbol{C}}_{ij}$ is computed as the mean predicted probability of class j for all trusted data points having true label of class i.

3.3 Training with Corrected Labels Let $\hat{\boldsymbol{C}}$ be the estimated noise corruption matrix. Using the method in [1], we increase the noise resilience by correcting the predictions of the classifier using $\hat{\boldsymbol{C}}$. Let $\hat{p}$ be the corrected predicted probabilities $\hat{p} = \hat{\boldsymbol{C}}^T p$, i.e. for data point $\boldsymbol{x}$: $\hat{p}(k|\boldsymbol{x}) = \sum_{i=1}^K \hat{\boldsymbol{C}}_{ik} p(i|\boldsymbol{x})$ for $k = 1, \ldots, K$. We enhance the regular cross entropy term. Applying the prediction correction to both terms

holds lower benefits. We evaluate this empirically with extensive experiments on datasets of text, i.e. Twitter in Figure 2(a), and images, i.e. CIFAR-100 and CIFAR-10 in §7. Experiment details can be found in §6. We consider different datasets, noise rates, noise types and fractions of trusted data. We see that in all cases, except one with a difference $< 0.3\%$, correcting only the cross entropy (*ce-only*) holds better results than correcting only the reverse cross entropy (*rce-only*) or correcting *both*. Focusing on Figure 2(a), *ce-only* improves accuracy by up to 5% and 8% for bimodal and symmetric noise, respectively. In case of CIFAR-10 and CIFAR-100 datasets the improvements are more pronounced with up to 11% and 50% respectively.

3.4 Golden Symmetric Loss Towards a more effective and robust learning we propose to leverage the estimated noise corruption matrix $\hat{\boldsymbol{C}}$ to tune the two loss terms based on the observed noise pattern. α and β can significantly impact the final model accuracy. Tuning these parameters is essential since various datasets affected by different noise patterns require different optimal values [20]. Again we show this behavior by training a 2-layer FC neural network on the Twitter dataset under eleven different (α, β) combinations and two noise patterns with 80% noise. Figure 2(b) reports for each noise pattern the evolution over the training epochs of the test accuracy for the best and worst (α, β)-pair. For bimodal noise even with a small number of trials, the impact of (α, β) ranges from an accuracy close to 60% all the way down to almost 0%. Moreover only few (two out of eleven) (α, β)-pairs hold accuracy close to 60%. For symmetric noise the tuning impact is lower (limited between 70% and 80%) but the best and worst (α, β)-pair differ from the bimodal noise case. This underlines both the importance and difficulty of tuning (α, β). Motivated by the high impact of α and β, we propose to dynamically weight the regular and reverse cross entropy terms. Let $A(\cdot)$ and $B(\cdot)$ be weighting functions mapping $\hat{\boldsymbol{C}} \to \mathbb{R}$ we define a new loss function:

$$\ell_{GSL} = A(\hat{\boldsymbol{C}})\, \ell_{ce} + B(\hat{\boldsymbol{C}})\, \ell_{rce} \tag{3.5}$$

We call this new loss function golden symmetric loss. $A(\cdot)$ and $B(\cdot)$ should capture not only the intensity of the noise pattern, but also the diversity of the noise pattern (see Figure 2(b)).

3.5 Determining Weights of Golden Symmetric Loss ($A(\cdot)$, $B(\cdot)$) In general the more intense and asymmetric the noise pattern, the lower the weight values should be. Since the final loss function learns from both dirty and clean data (see the next paragraph), lower values of α and β reduce the influence of dirty data over that of clean data. Hence, we design $A(\cdot)$ and $B(\cdot)$ to capture both noise intensity and diversity. The intensity is given by the noise rate $\varepsilon \in [0, \ldots, 1]$, i.e. one minus the average of the diagonal elements of $\hat{\boldsymbol{C}}$. The diversity is measured via Jain's fairness index $J(x_1, x_2, \ldots, x_n) \triangleq (\sum_{i=n}^{n} x_i)^2 / n \sum_{i=n}^{n} x_i^2$. We choose J because it bounds the diversity on a similar scale as ε between 1 (all equal, full symmetry) down to $1/n$ (highest asymmetry). We apply J on all the $K(K-1)$ noise, i.e. off the diagonal, elements of $\hat{\boldsymbol{C}}$:

$$J = \frac{(\sum_{i=1}^{K} \sum_{j=1, j\neq i}^{K} \hat{\boldsymbol{C}}_{ij})^2}{K(K-1) \sum_{i=1}^{K} \sum_{j=1, j\neq i}^{K} \hat{\boldsymbol{C}}_{ij}^2} \tag{3.6}$$

For symmetric noise $J = 1$, the more asymmetric the smaller J. Final weights proportional to J, ε.

3.6 Putting It All Together As a final step, to maximize the utility of the trusted data, we foster $\mathcal{D}$ as additional trusted training data for $f(\cdot)$. Since $\mathcal{D}$ contains the true labels y no prediction correction is applied. Hence, the overall loss function for data points from both $\mathcal{D}$ and $\tilde{\mathcal{D}}$ is:

for $x \in \tilde{\mathcal{D}}$,

$$\begin{aligned} l = &- A(\hat{\boldsymbol{C}}) \sum_{k=1}^{K} q(k|\boldsymbol{x}) \log(\sum_{i=1}^{K} \hat{\boldsymbol{C}}_{ik} p(i|\boldsymbol{x})) \\ &- B(\hat{\boldsymbol{C}}) \sum_{k=1}^{K} p(k|\boldsymbol{x}) \log q(k|\boldsymbol{x}) \end{aligned} \tag{3.7}$$

and for $x \in \mathcal{D}$,

$$l = -\sum_{k=1}^{K} q(k|\boldsymbol{x}) \log p(k|\boldsymbol{x}). \tag{3.8}$$

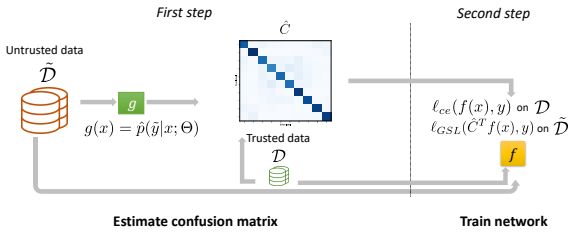


Figure 3: Training process of GSL divided into two steps.

Figure 3 summarises visually the training process divided into two main steps: (i) estimating noise corruption matrix through the first network g trained on untrusted dataset $\tilde{\mathcal{D}}$ and (ii) training classifier f on both untrusted

$\tilde{\mathcal{D}}$ and trusted $\mathcal{D}$ through the golden symmetric loss. The detailed steps of the algorithm can be found in the supplementary material.

4 Theoretical Analysis

We prove that the cross entropy loss with label correction is noise tolerant under the definition put forth by [19, 32] and extending prior art results. Let the risk of classifier f and loss function ℓ_{ce} under clean labels be $R(f) = \mathbb{E}_{\boldsymbol{x},y}[\ell_{ce}(f(\boldsymbol{x}), y)]$ and the *risk* under noise rate ε be $R_\varepsilon(f) = \mathbb{E}_{\boldsymbol{x},\tilde{y}}[\ell_{ce}(f(\boldsymbol{x}), \tilde{y})]$. $\mathbb{E}$ indicates the expectation taken over the random variables indicated as its subscripts. With prediction correction via $\boldsymbol{C}$, the risk becomes $R_\varepsilon(f, \boldsymbol{C}) = \mathbb{E}_{\boldsymbol{x},\tilde{y}}[\ell_{ce}(\boldsymbol{C}^T f(\boldsymbol{x}), \tilde{y})]$. Let f^* and f^*_ε be the global minimizers of $R(f)$ and $R_\varepsilon(f)$, respectively, and $\boldsymbol{C^*} = p(\tilde{y}|y)$ and $\hat{\boldsymbol{C}}$ be the true and estimated noise corruption matrices, respectively.

THEOREM 4.1. *In a multi-class classification problem, ℓ_{ce} with prediction correction is noise tolerant under symmetric label noise if the noise rate $\varepsilon < \frac{K-1}{K-\frac{\Delta\mathcal{A}}{\Delta R}}$, where $\Delta\mathcal{A} = \sum_{k=1}^{K} \ell_{ce}(\boldsymbol{C^{*T}} f(\boldsymbol{x}), k) - \sum_{k=1}^{K} \ell_{ce}(\hat{\boldsymbol{C}}^T f(\boldsymbol{x}), k)$, and ΔR is the difference of risk minimization between optimal classifier and f. And ℓ_{ce} with prediction correction is also noise tolerant under flip noise when noise rate $\varepsilon_{yk} \leq (1 + \frac{\Delta\mathcal{W}_y}{\Delta\mathcal{W}_k}) - \varepsilon_y(1 + \frac{\Delta\mathcal{W}_y}{\Delta\mathcal{W}_k})$ where ε_k and ε_{yk} are the correct and flipped class probabilities, respectively.*

The proof is based on the risk minimization framework aiming to show under which condition $R_\varepsilon(f^*, \boldsymbol{C}^*) - R_\varepsilon(f, \hat{\boldsymbol{C}}) \leq 0$, i.e. the loss function is robust to noise. The detailed steps of the proof can be found in the supplementary material. The condition $\varepsilon < \frac{K-1}{K-\frac{\Delta\mathcal{A}}{\Delta R}}$ is a generalization of the previous bound $\varepsilon < \frac{K-1}{K}$ by [19]. Without label correction $\Delta\mathcal{A} = 0$ which corresponds to the previous result. Label correction improves the robustness by allowing higher noise rates, i.e. with label correction $\frac{\Delta\mathcal{A}}{\Delta R} \geq 0$ and hence $\frac{K-1}{K} \leq \frac{K-1}{K-\frac{\Delta\mathcal{A}}{\Delta R}}$. Similar observations hold for flip noise bound.

5 Experimental Setup

5.1 Dataset, Architecture and Parameters We consider two types of datasets: vision and text analysis. For vision, we use convolution neural networks (CNN) to classify CIFAR-10 and CIFAR-100 with injected label noise and Clothing1M as real world noisy dataset. For text, we use fully connected neural networks to classify noisy Twitter and Stanford Sentiment Treebank (SST). In principle, we use the same network architecture on all comparative approaches across different noise resilience techniques. In addition, we test the original network from the respective papers too and report the best results among the two.

- **CIFAR-10 [33]:** It contains 60K images classified into 10 classes: 50K as a training set and 10K as validation set. We use the architecture of Wide-ResNet by [34] with depth 28 and a widening factor 10 and train it with SGD with Nesterov momentum and a cosine learning rate schedule [35]. For GSL, we first train g for 75 epochs to obtain the noise corruption matrix. Then we train f for 120 epochs.
- **CIFAR-100 [33]:** It contains 60K images classified into 100 classes: 50K as training set and 10K as the validation set. We use the same Wide-ResNet architecture used for CIFAR-10. For GSL, we train the g and f networks for 75 and 200 epochs, respectively.
- **Clothing1M [7]:** This is a real world dataset with label noise. It includes clothing images scrapped from the Internet classified into 14 categories. We resize and crop each image to 224×224 pixels. This dataset contains 1 million noisy labeled samples that we use for training as our untrusted data. Besides, it consists of 57K human-annotated images, which we take 47K images as our trusted examples and 10K images for testing. These two sets have both given (scrapped) and true (human-checked) labels. We use ResNet-50 pretrained with ImageNet and further train for 10 epochs with batch size 32, SGD optimizer, momentum 0.9, weight decay 10^{-3}, and learning rate 10^{-3} which is divided by 10 after 5 epochs.
- **Twitter [36]:** The Twitter dataset includes 1,827 tweets annotated with 25 POS tags split in 1000 tweets as training set, 327 tweets as development set and 500 tweets as test set. We add development set to training set, and consider it as a training set. We use a 2-layer fully connected network with 256 hidden neurons each and GELU nonlinearity as activation function. We train g with Adam for 15 epochs with batch size 64 and learning rate of 0.001. We train f for 25 epochs. To regularize all linear output layer, we use ℓ_2 weight decay with $\lambda = 5 \times 10^{-5}$.
- **Stanford Sentiment Treebank [37]:** The SST dataset includes single sentence movie reviews. We use the 2-class version, including 6911 reviews in the training set, a development set with 872 reviews, and 1821 reviews in the test set. We augment the training set by using development set. We learn 100-dimensional word vectors from scratch for a vocab

size of 10000. We train a word-averaging model with an affine output layer using Adam optimizer for 5 epochs for network g and 10 epochs for network f. The batch size and learning rate are 50 and 0.001, respectively. To regularize all linear output layer, we use ℓ_2 weight decay with $\lambda = 1 \times 10^{-4}$.

5.2 Noise Corruption We consider symmetric noise and two different asymmetric noises, namely flip and bimodal. Symmetric noise corrupts the true label into a random other labels with equal probability based on the noise rate. The flip noise is generated by flipping the original label to a paired other class with a specific probability. The bimodal noise imitates targeted adversarial attacks [2]. Specifically, the true labels are corrupted into two neighborhoods centered on two targeted classes, each of which follows truncated normal distribution, $\mathcal{N}^T(\mu, \sigma, a, b)$. μ specifies the target and σ controls the spread. a and b simply define the class label boundaries. For CIFAR-10 we target class 3 and 7, for CIFAR-100 class 30 and 70, for Twitter class 6 and 18, and for SST class 0 and 1. Instead, Clothing1M is already affected by real world label noise and left untouched. More details are provided in the supplementary material.

6 Evaluation

In this section, we empirically compare GSL against state of the art noise resilient networks on noisy vision and text data. We aim to show the effectiveness of GSL via testing accuracy on diverse and challenging noise patterns. Our target evaluation metric is the accuracy achieved on the clean testing set, i.e. not affected by noise.

6.1 Vision Analysis We compare GSL against ten noise resilient networks from the state of the art: GLC [6], SCL [20], Forward [1], Bootstrap [30], Co-teaching+ [13], DivideMix [31], gForward, sgForward, TMatrix and sTMatrix. As the proposed loss of golden symmetric cross entropy is general and can be combined with different resilient networks, we hence use following four variations of loss correction and symmetric cross entropy on the existing work:

- Forward gold (gForward): we replace the estimation of the corruption matrix by the identity matrix on trusted samples and apply loss correction through the matrix.

- True corruption matrix (TMatrix): we use the true corruption matrix and apply loss correction through it.

- Forward gold with symmetric cross entropy (sgForward): we extend the corrected loss of gForward to the corrected symmetric cross entropy as in the GSL.

- True corruption with symmetric cross entropy (sTMatrix): we apply golden symmetric cross entropy and the true corruption matrix instead of the estimated matrix.

For training GSL, Co-teaching+, sgForward, gForward, TMatrix, sTMatrix, DivideMix and GLC, we use PyTorch v1.4.0. For all other methods, we use Keras v2.2.4 and Tensorflow v1.13.0. All experiments run on Alienware Aurora R11 equipped with an NVIDIA GeForce RTX 2080 Ti, 32 GB RAM, and Core i9 CPU @ 3.70 GHz.

We assume 10% of trusted data is available for GSL, GLC, gForward and sgForward. Table 1 summarizes the testing accuracy for all combinations of noise patterns and comparative approaches.

For CIFAR-10, we report the average and standard deviation across three runs in Table 1. GSL achieves the highest accuracy among all resilient networks except for flip noise with 30% noise rate. DivideMix, sTMatrix and sgForward are the closest rivals to GSL. GSL and sgForward both use the same mechanism in the loss function. Besides, GSL has 2 to 8% higher accuracy than GLC, demonstrating the benefit of introducing symmetric cross-entropy, especially in high noise rates. In terms of comparison between GSL and SCL, the accuracy difference is even more visible, implying the benefit of using corruption matrix to assign weights on two terms in symmetric cross-entropy. We note that SCL uses an 8-layer CNN with 6 convolutional layers followed by 2 fully connected layers instead of a Wide ResNet because of the superior results. SCL performs particularly worse in 60% bimodal noise because this is a more challenging pattern and has no access to the corruption matrix. Also, we achieve higher accuracy than DivideMix which is one of the accurate state-of-the-art. Moreover, our method can still obtain 11 to 30% higher test accuracy than Co-teaching+ that uses two deep networks concurrently.

CIFAR-100 is more challenging than CIFAR-10 due to the larger number of classes and results are summarized over three runs in Table 1. GSL achieves the highest accuracy except for flip noise with 30% rate, and same as CIFAR-10, sTMatrix, DivideMix, and sgForward are the closest competitors. Although for flip noise with 30% rate sgForward performs better than GSL, the improvement of GSL is more significant than sgForward compared to the CIFAR-10 dataset. The largest difference (more than 2%) in accuracy

Table 1: Vision analysis: test accuracy(%) of real-world noisy Clothing1M, and CIFAR10/CIFAR100 corrupted with 30% and 60% noise for different noise resilient networks. Best results in bold.

CIFAR-10

Noise Rate	Noise Pattern	GSL	GLC	SCL	Forward	Bootstrap	sgForward	Co-teaching+	DivideMix	gForward	TMatrix	sTMatrix
30%	Sym.	**92.90** ± 0.24	89.94 ± 0.36	83.50 ± 0.28	74.28 ± 0.20	75.62 ± 0.15	90.81 ± 0.22	76.70 ± 0.72	91.68 ± 0.28	79.76 ± 0.92	90.66 ± 0.19	91.09 ± 0.36
30%	Bimodal	**92.81** ± 0.18	90.18 ± 0.91	83.06 ± 0.21	73.42 ± 0.54	75.69 ± 0.12	91.10 ± 0.45	75.21 ± 0.54	84.13 ± 0.18	78.45 ± 0.49	89.95 ± 0.31	90.59 ± 0.24
30%	Flip	90.30 ± 0.36	**91.15** ± 0.17	81.53 ± 0.33	78.72 ± 0.19	78.51 ± 0.38	90.54 ± 0.42	79.83 ± 0.78	86.01 ± 0.43	81.18 ± 0.51	88.79 ± 0.71	89.29 ± 0.73
60%	Sym.	**89.10** ± 0.19	82.24 ± 0.59	72.71 ± 0.86	53.48 ± 0.79	57.56 ± 1.86	88.02 ± 0.56	63.33 ± 0.93	88.27 ± 0.76	60.62 ± 0.61	87.31 ± 0.14	88.07 ± 0.53
60%	Bimodal	**87.75** ± 0.24	84.98 ± 0.16	60.76 ± 0.82	47.49 ± 0.69	48.18 ± 1.01	86.29 ± 0.52	57.82 ± 0.73	81.45 ± 0.37	58.93 ± 0.46	84.33 ± 0.62	86.79 ± 0.21
60%	Flip	**86.23** ± 1.10	80.40 ± 0.32	55.84 ± 0.70	59.99 ± 0.47	59.66 ± 0.45	82.19 ± 0.43	65.31 ± 0.36	79.76 ± 0.67	62.04 ± 0.63	81.88 ± 0.37	84.91 ± 0.37

CIFAR-100

Noise Rate	Noise Pattern	GSL	GLC	SCL	Forward	Bootstrap	sgForward	Co-teaching+	DivideMix	gForward	TMatrix	sTMatrix
30%	Sym.	**75.80** ± 0.12	61.81 ± 1.19	58.01 ± 0.71	42.33 ± 1.34	41.51 ± 1.54	72.31 ± 0.77	54.04 ± 0.33	72.73 ± 0.22	52.64 ± 0.73	70.42 ± 0.71	73.04 ± 0.69
30%	Bimodal	**76.25** ± 0.35	61.77 ± 0.91	46.88 ± 0.63	45.22 ± 0.13	42.14 ± 0.38	73.65 ± 0.29	55.42 ± 0.65	74.21 ± 0.18	54.69 ± 0.82	72.07 ± 0.32	74.41 ± 0.22
30%	Flip	75.80 ± 0.21	66.55 ± 0.52	55.46 ± 0.47	54.92 ± 0.25	54.44 ± 0.59	**75.83** ± 0.42	58.46 ± 0.61	74.16 ± 0.39	58.32 ± 0.20	73.11 ± 0.14	75.15 ± 0.43
60%	Sym.	**68.49** ± 0.16	52.23 ± 0.85	29.00 ± 0.54	18.56 ± 1.11	16.22 ± 0.81	66.32 ± 0.79	38.15 ± 0.94	66.81 ± 0.66	39.32 ± 0.33	63.48 ± 0.22	66.87 ± 0.44
60%	Bimodal	**65.39** ± 0.48	50.33 ± 1.05	29.12 ± 0.77	18.79 ± 0.82	10.32 ± 0.63	63.03 ± 0.66	34.09 ± 0.15	64.88 ± 0.38	41.65 ± 0.79	63.84 ± 0.53	64.29 ± 0.23
60%	Flip	**69.60** ± 0.42	66.58 ± 0.43	41.37 ± 0.66	40.18 ± 1.34	37.27 ± 0.75	67.42 ± 0.38	40.68 ± 0.36	65.09 ± 0.86	42.77 ± 0.14	65.21 ± 0.66	67.38 ± 0.44

Clothing1M

Noise	GSL	GLC	SCL	Forward	Bootstrap	sgForward	Co-teaching+	DivideMix	gForward	TMatrix	sTMatrix
Real World	**74.86**	73.91	70.78	70.04	67.87	73.96	70.33	74.29	70.95	72.04	72.41

Table 2: Text analysis: average accuracy (%) of variants combining loss correction and symmetric cross entropy. Results averaged across entire range of noise rates $[0, 100]$. Best accuracy in bold.

	Noise Pattern	Percent Trusted	GSL	GLC	gForward	TMatrix	sgForward	sTMatrix	Forward	SCL	Bootstrap	Co-teaching+	DivideMix
Twitter	Sym.	1	**79.30** ± 0.11	65.41 ± 0.90	53.21 ± 0.17	76.61 ± 0.37	78.39 ± 0.18	78.24 ± 0.24	52.25 ± 0.25	62.77 ± 0.63	50.59 ± 0.12	65.79 ± 0.22	76.95 ± 0.76
	Sym.	5	**81.94** ± 0.29	77.20 ± 0.17	59.61 ± 0.44	79.63 ± 0.33	81.20 ± 0.16	81.33 ± 0.17	59.07 ± 0.54	63.53 ± 0.31	52.04 ± 0.30	67.67 ± 0.34	78.73 ± 0.42
	Bimodal	1	75.92 ± 0.29	67.15 ± 0.28	52.53 ± 0.19	**77.64** ± 0.56	75.49 ± 0.34	76.73 ± 0.39	50.13 ± 0.40	62.31 ± 0.24	49.11 ± 0.25	63.89 ± 0.54	75.06 ± 0.27
	Bimodal	5	**84.35** ± 0.39	78.45 ± 0.37	60.63 ± 0.28	80.58 ± 0.32	80.41 ± 0.88	80.73 ± 0.19	54.64 ± 0.72	66.87 ± 0.31	53.87 ± 0.38	68.94 ± 0.32	81.95 ± 0.62
	Flip	1	82.75 ± 0.63	83.13 ± 0.24	39.52 ± 0.22	**86.13** ± 0.31	73.89 ± 0.41	73.28 ± 0.18	48.21 ± 0.31	60.63 ± 0.15	48.87 ± 0.28	62.66 ± 0.29	84.66 ± 0.26
	Flip	5	84.75 ± 0.31	85.49 ± 0.38	48.42 ± 0.61	**87.04** ± 0.21	79.48 ± 0.36	80.20 ± 0.18	53.87 ± 0.65	64.74 ± 0.39	51.88 ± 0.18	66.19 ± 0.29	86.29 ± 0.42
SST	Sym.	0.1	**75.18** ± 0.55	73.47 ± 0.28	72.15 ± 0.29	73.55 ± 0.29	72.22 ± 0.09	73.66 ± 0.49	70.13 ± 0.31	71.36 ± 0.26	70.03 ± 0.42	71.84 ± 0.33	74.07 ± 0.17
	Sym.	1	**75.96** ± 0.46	72.62 ± 0.28	73.47 ± 0.25	75.48 ± 0.36	72.93 ± 0.19	75.42 ± 0.34	72.52 ± 0.27	72.86 ± 0.39	71.31 ± 0.22	72.13 ± 0.29	74.93 ± 0.29
	Bimodal	0.1	**74.97** ± 0.24	74.70 ± 0.31	72.75 ± 0.14	74.19 ± 0.44	72.63 ± 0.51	74.16 ± 0.38	70.02 ± 0.18	71.22 ± 0.16	70.21 ± 0.21	72.25 ± 0.15	73.93 ± 0.31
	Bimodal	1	**74.88** ± 0.41	74.53 ± 0.32	72.10 ± 0.13	74.34 ± 0.29	71.79 ± 0.42	73.60 ± 0.30	72.67 ± 0.29	72.13 ± 0.41	71.38 ± 0.19	72.93 ± 0.31	74.01 ± 0.16
	Flip	0.1	**75.38** ± 0.29	74.07 ± 0.25	49.40 ± 0.51	74.83 ± 0.22	49.50 ± 0.34	74.81 ± 0.16	70.79 ± 0.54	72.52 ± 0.12	70.79 ± 0.41	72.14 ± 0.27	74.54 ± 0.30
	Flip	1	**76.59** ± 0.14	74.51 ± 0.32	50.21 ± 0.43	76.33 ± 0.13	49.81 ± 0.23	75.49 ± 0.43	73.04 ± 0.17	73.76 ± 0.51	71.78 ± 0.14	72.83 ± 0.10	74.99 ± 0.25

between the GSL and sgForward methods is with bimodal noise, and between GSL and sTMatrix is with flip noise. In case of 60% symmetric noise, GSL achieves the accuracy of 68%, whereas GLC and SCL trail far behind. Moreover, given the difficulty of training a robust classifier for CIFAR-100 with 60% label noise, it is worth mentioning that SCL can achieve similar performance as GLC that is given 10% of trusted data in case of 30% symmetric noise. This also indicates the effectiveness of symmetric cross entropy in learning hard classes even without trusted data. However, when facing extremely noisy labels and patterns, the small amount of trusted data can greatly improve the robustness of the classifier but not necessarily the symmetric cross entropy.

Seen from the high accuracy compared to GLC, SCL, gForward and sgForward, GSL effectively uses the trusted data to correct symmetric cross entropy loss and improve the learning on the hard classes. GSL performs slightly better with symmetric noise than with bimodal and flip noise that is more challenging for CIFAR-10. In the CIFAR-100, GSL works better on the asymmetric noise rather than symmetric.

For Clothing1M dataset, as shown in Table 1, GSL obtains the highest test accuracy compared to other methods. Same as CIFAR-10 and CIFAR-100, DivideMix achieves a relatively good performance. The difference between GSL and SCL comes from the effectiveness of corruption matrix that makes the regular cross entropy robust.

6.2 Text Analysis We evaluate GSL on text datasets of Twitter and SST, against resilient networks that leverage corruption matrix, namely GLC and Forward. Both GSL and GLC use the trusted data for estimating the corruption matrix, wheres the original Forward [1] relies solely on the noisy data.

We extensively evaluate GSL, GLC gForward, TMatrix, sgForward, SCL, Forward, Bootstrap, Co-teaching+, DivideMix and sTMatrix on Twitter and SST, with label corruption ranging from 0% to 100%. We also vary the percentage of trusted data among 1% and 5%. We summarize the average accuracy across 11 noise rates and three runs in Table 2.

6.2.1 Twitter As shown in Table 2, GSL consistently achieves the highest average accuracy in most cases. Compared to GLC, GSL has significant higher accuracy for Twitter corrupted with symmetric and bimodal noises, but the difference diminishes with increasing amounts of trusted data. When the percent of trusted data is low, say, 1%, GLC is unable to estimate the corruption matrix accurately nor to correct the loss, seen by the difference between GLC and TMatrix.

6.2.2 SST Here, the classification involves only two classes and turns out to be less challenging than the Twitter case. The results in Table 2 show that the difference among the different comparative approaches considered is smaller than for Twitter. For instance, though GSL consistently achieves the best average accuracy in almost all cases, the difference between GSL and GLC is around 1-3%. Again, we see that GSL visibly outperforms GLC on low amounts of trusted data because of using cross entropy and the difference among them becomes limited. We note that TMatrix and gForward collapse under Flip noise. We plot an extension of Table 2 for analysis on text datasets for varying noise rates in the supplemental material.

7 Discussion

Here we present the extensive results of our empirical evaluation on training with corrected labels for the vision datasets in Table 3 and Table 4, respectively. This complements the results presented in §3. We compare the impact of correcting labels only on the cross entropy term (*ce only*), only on the reverse cross entropy term (*rce only*), or *both*. Table 3 shows the achieved accuracy for CIFAR-100, under two noise rates, 30% and 60%, three different noise types, symmetric, bimodal and flip, and three fractions of trusted data, 5%, 10% and 15%. For each noise scenario the best case is highlighted in bold.

Table 3: Accuracy (%) of different gold fraction on CIFAR-100

Noise rate = 30%									
Label correction	Bimodal			Symmetric			Flip		
	5%	10%	15%	5%	10%	15%	5%	10%	15%
ce only	**71.66**	**73.90**	**75.20**	**71.69**	**74.24**	**75.11**	**74.78**	**75.46**	**77.01**
rce only	25.90	61.44	67.15	26.15	61.19	67.14	23.75	59.68	65.33
both	23.37	57.52	64.30	23.58	57.36	63.76	19.74	54.50	61.20
Noise rate = 60%									
Label correction	Bimodal			Symmetric			Flip		
	5%	10%	15%	5%	10%	15%	5%	10%	15%
ce only	**58.42**	66.96	**69.41**	**55.22**	**66.87**	**69.46**	**65.20**	**68.33**	**70.41**
rce only	54.35	**67.24**	69.41	24.90	58.72	64.39	14.43	46.09	68.75
both	24.35	55.18	61.19	25.62	55.16	61.46	12.46	36.76	50.83

ce only achieves the highest accuracy in all cases except one. Under 60% bimodal noise on CIFAR-100 with 10% trusted data *rce only* is slightly better by 0.28 percent points. More in general, *rce only* typically performs second best and *both* achieves the worst accuracy. Focusing on *ce only* over the other two, the gain tends to increase with the difficulty of the noise scenarios, i.e. with higher number of classes, higher noise rates and less trusted data. *ce only* outperforms the other two by up to 51.03 percent points for CIFAR-100. Same as CIFAR-100 results, *ce only* outperforms the other two by up to 11.74 percent points for CIFAR-10 in Table 4.

Table 4: Accuracy (%) of different gold fraction on CIFAR-10

Noise rate = 30%									
Label correction	Bimodal			Symmetric			Flip		
	5%	10%	15%	5%	10%	15%	5%	10%	15%
ce only	**90.06**	**91.50**	**92.53**	**90.17**	**91.70**	**92.50**	**89.96**	**91.27**	**92.58**
rce only	84.67	88.46	89.30	84.82	88.11	89.98	81.70	86.79	88.40
both	83.25	87.45	89.30	83.43	87.70	89.46	78.00	85.42	88.21
Noise rate = 60%									
Label correction	Bimodal			Symmetric			Flip		
	5%	10%	15%	5%	10%	15%	5%	10%	15%
ce only	**83.06**	**88.43**	**90.52**	**85.79**	**89.19**	**90.19**	**80.03**	**82.64**	**85.80**
rce only	81.73	86.86	89.07	81.54	86.63	88.98	68.29	80.22	84.35
both	79.83	85.79	88.63	80.27	86.22	88.53	63.63	82.19	83.18

8 Conclusion

To enhance the robustness of deep models against by label noise, we propose GSL that features on correcting the symmetric cross entropy loss by the noise corruption matrix. GSL uses a small fraction of trusted data to accurately estimate the corruption matrix, and further determine the weights applied on regular and reverse cross entropy. GSL learns deep networks from trusted samples through regular cross entropy and from untrusted noisy samples through golden symmetric cross entropy. We prove that the cross entropy corrected by the corruption matrix is noise robust. To adapt to noise patterns of dataset, we heuristically set the weights of golden symmetric loss based on the corruption matrix. We extensively evaluate GSL on vision and text analysis under diversified noise rates and patterns. Evaluation results show that GSL can achieve a remarkable accuracy improvement, i.e., from 2 to 18% on CIFAR benchmarks and real world noisy data, compared to methods that either correct loss or leverage symmetric cross entropy.

9 Acknowledgements

This work is supported by the TEXTAROSSA project (G.A. n. 956831), as part of the EuroHPC initiative.

This work has been partially supported by European Commission and the Italian Ministry of Economic Development (MISE) under the EuroHPC TEXTAROSSA project (G.A. 956831).

References

[1] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label

noise: A loss correction approach," in *CVPR*, pp. 1944–1952, 2017.
[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *ICLR*, 2015.
[3] L. Jiang, Z. Zhou, T. Leung, L. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *ICML*, pp. 2309–2318, 2018.
[4] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *ICLR*, 2017.
[5] Y. Li, J. Yang, Y. Song, L. Cao, J. Luo, and L. Li, "Learning from noisy labels with distillation," in *ICCV*, pp. 1928–1936, 2017.
[6] D. Hendrycks, M. Mazeika, D. Wilson, and K. Gimpel, "Using trusted data to train deep networks on labels corrupted by severe noise," in *NIPS*, pp. 10456–10465, 2018.
[7] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *CVPR*, pp. 2691–2699, 2015.
[8] A. Ghiassi, R. Birke, and L. Y. Chen, "Trustnet: Learning from trusted data against (a) symmetric label noise," in *BDCAT*, pp. 52–62, 2021.
[9] Y. Wang, W. Liu, X. Ma, J. Bailey, H. Zha, L. Song, and S.-T. Xia, "Iterative learning with open-set noisy labels," in *CVPR*, pp. 8688–8696, 2018.
[10] P. Chen, B. Liao, G. Chen, and S. Zhang, "Understanding and utilizing deep neural networks trained with noisy labels," in *ICML*, pp. 1062–1070, 2019.
[11] A. Vahdat, "Toward robustness against label noise in training deep discriminative neural networks," in *NIPS*, pp. 5596–5605, 2017.
[12] B. Han, Q. Yao, X. Yu, G. Niu, M. Xu, W. Hu, I. Tsang, and M. Sugiyama, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *NIPS*, pp. 8527–8537, 2018.
[13] X. Yu, B. Han, J. Yao, G. Niu, I. W. Tsang, and M. Sugiyama, "How does disagreement help generalization against label corruption?," in *ICML*, pp. 7164–7173, 2019.
[14] A. Ghiassi, R. Birke, R. Han, and L. Y. Chen, "Labelnet: Recovering noisy labels," in *IJCNN*, pp. 1–8, 2021.
[15] A. Ghiassi, T. Younesian, Z. Zhao, R. Birke, V. Schiavoni, and L. Y. Chen, "Robust (deep) learning framework against dirty labels and beyond," in *IEEE TPS-ISA*, pp. 236–244, 2019.
[16] Z. Jiang, K. Zhou, Z. Liu, L. Li, R. Chen, S.-H. Choi, and X. Hu, "An information fusion approach to learning with instance-dependent label noise," in *ICLR*, 2021.
[17] Z. Zhu, T. Liu, and Y. Liu, "A second-order approach to learning with instance-dependent label noise," in *CVPR*, pp. 10113–10123, 2021.
[18] N. Konstantinov and C. Lampert, "Robust learning from untrusted sources," in *ICML*, vol. 97, pp. 3488–3498, 2019.
[19] A. Ghosh, H. Kumar, and P. S. Sastry, "Robust loss functions under label noise for deep neural networks," in *AAAI*, pp. 1919–1925, 2017.
[20] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *ICCV*, pp. 322–330, 2019.
[21] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *ICML*, pp. 4331–4340, 2018.
[22] X. Ma, Y. Wang, M. E. Houle, S. Zhou, S. M. Erfani, S. Xia, S. N. R. Wijewickrema, and J. Bailey, "Dimensionality-driven learning with noisy labels," in *ICML*, pp. 3361–3370, 2018.
[23] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *NIPS*, pp. 8792–8802, 2018.
[24] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng, "Meta-weight-net: Learning an explicit mapping for sample weighting," in *NIPS*, pp. 1919–1930, 2019.
[25] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," *ICML*, 2018.
[26] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, "Joint optimization framework for learning with noisy labels," in *CVPR*, pp. 5552–5560, 2018.
[27] C. Hong, A. Ghiassi, Y. Zhou, R. Birke, and L. Y. Chen, "Online label aggregation: A variational bayesian approach," in *WWW*, pp. 1904–1915, 2021.
[28] A. Veit, N. Alldrin, G. Chechik, I. Krasin, A. Gupta, and S. J. Belongie, "Learning from noisy large-scale datasets with minimal supervision," in *CVPR*, pp. 6575–6583, 2017.
[29] B. Han, J. Yao, G. Niu, M. Zhou, I. W. Tsang, Y. Zhang, and M. Sugiyama, "Masking: A new perspective of noisy supervision," in *NIPS*, pp. 5841–5851, 2018.
[30] S. E. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *ICLR 2015*.
[31] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," in *ICLR*, 2020.
[32] N. Manwani and P. S. Sastry, "Noise tolerance under risk minimization," *IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1146–1151, 2013.
[33] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10/100 (Canadian Institute for Advanced Research)," 2009.
[34] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *BMVC*, 2016.
[35] I. Loshchilov and F. Hutter, "SGDR: stochastic gradient descent with warm restarts," in *ICLR*, 2017.
[36] K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan, and N. A. Smith, "Part-of-speech tagging for twitter: Annotation, features, and experiments," in *ACL*, pp. 42–47, 2011.
[37] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *EMNLP*, pp. 1631–1642, 2013.