Normalized Loss Functions for Deep Learning with Noisy Labels

Xingjun Ma [* 1] **Hanxun Huang** [* 1] **Yisen Wang** [2] **Simone Romano** **Sarah Erfani** [1] **James Bailey** [1]

Abstract

Robust loss functions are essential for training accurate deep neural networks (DNNs) in the presence of noisy (incorrect) labels. It has been shown that the commonly used Cross Entropy (CE) loss is not robust to noisy labels. Whilst new loss functions have been designed, they are only partially robust. In this paper, we theoretically show by applying a simple normalization that: *any loss can be made robust to noisy labels*. However, in practice, simply being robust is not sufficient for a loss function to train accurate DNNs. By investigating several robust loss functions, we find that they suffer from a problem of *underfitting*. To address this, we propose a framework to build robust loss functions called *Active Passive Loss* (APL). APL combines two robust loss functions that mutually boost each other. Experiments on benchmark datasets demonstrate that the family of new loss functions created by our APL framework can consistently outperform state-of-the-art methods by large margins, especially under large noise rates such as 60% or 80% incorrect labels.

1. Introduction

Training accurate deep neural networks (DNNs) in the presence of noisy (incorrect) labels is of great practical importance. Different approaches have been proposed for robust learning with noisy labels. This includes 1) label correction methods that aim to identify and correct wrong labels (Xiao et al., 2015; Vahdat, 2017; Veit et al., 2017; Li et al., 2017b); 2) loss correction methods that correct the loss function based on an estimated noise transition matrix (Sukhbaatar et al., 2014; Reed et al., 2014; Patrini et al., 2017; Han et al., 2018a); 3) refined training strategies that modify the training procedure to be more adaptive to incorrect labels (Jiang et al., 2018; Wang et al., 2018; Tanaka et al., 2018; Ma et al.,

2018; Han et al., 2018b); and 4) robust loss functions that are inherently tolerant to noisy labels (Ghosh et al., 2017; Zhang & Sabuncu, 2018; Wang et al., 2019c). Compared to the first three approaches that may suffer from inaccurate noise estimation or involve sophisticated training procedure modifications, robust loss functions provide a simpler solution, which is also the main focus of this paper.

It has been theoretically shown that some loss functions such as Mean Absolute Error (MAE) are robust to label noise, while others are not, which unfortunately includes the commonly used Cross Entropy (CE) loss. This has motivated a body of work to design new loss functions that are inherently robust to noisy labels. For example, Generalized Cross Entropy (GCE) (Zhang & Sabuncu, 2018) was proposed to improve the robustness of CE against noisy labels. GCE can be seen as a generalized mixture of CE and MAE, and is only robust when reduced to the MAE loss. Recently, a Symmetric Cross Entropy (SCE) (Wang et al., 2019c) loss was suggested as a robustly boosted version of CE. SCE combines the CE loss with a Reverse Cross Entropy (RCE) loss, and only the RCE term is robust. Whilst these loss functions have demonstrated improved robustness, theoretically, they are only partially robust to noisy labels.

Different from previous works, in this paper, we theoretically show that any loss can be made robust to noisy labels, and all is needed is a simple normalization. However, in practice, simply being robust is not enough for a loss function to train accurate DNNs. By investigating several robust loss functions, we find that they all suffer from an underfitting problem. Inspired by recent developments in this field, we propose to characterize existing loss functions into two types: 1) "Active" loss, which only explicitly maximizes the probability of being in the labeled class, and 2) "Passive" loss, which also explicitly minimizes the probabilities of being in other classes. Based on this characterization, we further propose a novel framework to build a new set of robust loss functions called *Active Passive Losses* (APLs). We show that under this framework, existing loss functions can be reworked to achieve the state-of-the-art for training DNNs with noisy labels. Our key contributions are:

- We provide new theoretical insights into robust loss functions demonstrating that a simple normalization can make any loss function robust to noisy labels.

---
[*]Equal contribution [1]The University of Melbourne, Australia [2]Shanghai Jiao Tong University, China. Correspondence to: Yisen Wang <eewangyisen@gmail.com>.

- We identify that existing robust loss functions suffer from an underfitting problem. To address this, we propose a generic framework *Active Passive Loss* (APL) to build new loss functions with theoretically guaranteed robustness and sufficient learning properties.

- We empirically demonstrate that the family of new loss functions created following our *APL* framework can outperform the state-of-the-art methods by considerable margins, especially under large noise rates of 60% or 80%.

2. Related Work

We briefly review existing approaches for robust learning with noisy labels.

1) Label correction methods. The idea of label correction is to improve the quality of the raw labels, possibly correcting wrong labels into correct ones. One common approach is to apply corrections via a clean label inference step using complex noise models characterized by directed graphical models (Xiao et al., 2015), conditional random fields (Vahdat, 2017), neural networks (Lee et al., 2017; Veit et al., 2017) or knowledge graphs (Li et al., 2017b). These methods require support from extra clean data or a potentially expensive detection process to estimate the noise model.

2) Loss correction methods. This approach improves robustness by modifying the loss function during training, based on label-dependent weights (Natarajan et al., 2013) or an estimated noise transition matrix that defines the probability of mislabeling one class with another (Han et al., 2018a). Backward and Forward (Patrini et al., 2017) are two noise transition matrix based loss correction methods. Work in (Goldberger & Ben-Reuven, 2017; Sukhbaatar et al., 2014) augments the correction architecture by adding a linear layer on top of the neural network. Bootstrap (Reed et al., 2014) uses a combination of raw labels and their predicted labels. Label Smoothing Regularization (LSR) (Szegedy et al., 2016; Pereyra et al., 2017) uses soft labels in place of one-hot labels to alleviate overfitting to noisy labels. Loss correction methods are sensitive to the noise transition matrix. Given that ground-truth is not always available, this matrix is typically difficult to estimate.

3) Refined training strategies. This direction designs adaptive training strategies that are more robust to noisy labels. MentorNet (Jiang et al., 2018; Yu et al., 2019) supervises the training of a StudentNet by a learned sample weighting scheme in favor of probably correct labels. SeCoST extends MentorNet to a cascade of student-teacher pairs via a knowledge transfer method (Kumar & Ithapu, 2019). Decoupling training strategy (Malach & Shalev-Shwartz, 2017) trains two networks simultaneously, and parameters are updated when their predictions disagree. Co-teaching (Han et al., 2018b) allows one network learn from the other network's

most confident samples. These studies all require an auxiliary network for sample weighting or learning supervision. D2L (Ma et al., 2018) uses subspace dimensionality adapted labels for learning, paired with a training process monitor. The joint optimization framework (Tanaka et al., 2018) updates DNN parameters and labels alternately. Kim et al. (2019) use complementary labels to mitigate overfitting to original labels. Xu et al. (2019) introduce a Determinant-based Mutual Information (DMI) loss for robust fine-tuning of a CE pre-trained model. These methods either rely on complex interventions into the learning process which are hard to adapt and tune, or are sensitive to hyperparameters like training epochs and learning rate.

4) Robust loss functions. Compared to the above three types of methods, robust loss functions are a simpler and arguably more generic solution for robust learning. Previous work has theoretically proved that some loss functions such as Mean Absolute Error (MAE) are robust to noisy labels, while others like the commonly used Cross Entropy (CE) loss are not (Ghosh et al., 2017). However, training with MAE has been found very challenging due to slow convergence caused by gradient saturation (Zhang & Sabuncu, 2018). The Generalized Cross Entropy (GCE) loss (Zhang & Sabuncu, 2018) applies a Box-Cox transformation to probabilities (power law function of probability with exponent $\rho \in (0, 1]$) which can behave like a generalized mixture of MAE and CE. Recently, Wang et al. (2019c) proposed the Symmetric Cross Entropy (SCE) which combines a Reverse Cross Entropy (RCE) together with the CE loss. Both GCE and SCE are only partially robust to noisy labels. For example, GCE is only robust when it reduces to the MAE loss with $\rho = 1$. For SCE, only its RCE term is robust. Empirically (rather than theoretically) justified approaches that directly modify the magnitude of the loss gradients are also an active line of research (Wang et al., 2019a;b).

In this paper, we theoretically prove that, with simple normalization, any loss can be made robust to noisy labels. This new theoretical insight can serve as a basic principle for designing new robust loss functions. It also can reshape the design of new loss functions towards other properties rather than robustness.

3. Any Loss can be Robust to Noisy Labels

We next introduce some background knowledge about robust classification with noisy labels, then propose a simple but theoretically sound normalization method that can be applied to any loss function to make it robust to noisy labels.

3.1. Preliminaries

Given a K-class dataset with noisy labels as $\mathcal{D} = \{(\boldsymbol{x}, y)^{(i)}\}_{i=1}^n$, with $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d$ denoting a sample and

$y \in \mathcal{Y} = \{1, \cdots, K\}$ its annotated label (possibly incorrect). We denote the distribution over different labels for sample $\boldsymbol{x}$ by $\boldsymbol{q}(k|\boldsymbol{x})$, and $\sum_{k=1}^{K} \boldsymbol{q}(k|\boldsymbol{x}) = 1$. In this paper, we focus on the common case where there is only one single label y for $\boldsymbol{x}$: i.e. $\boldsymbol{q}(y|\boldsymbol{x}) = 1$ and $\boldsymbol{q}(k \neq y|\boldsymbol{x}) = 0$. In this case, $\boldsymbol{q}$ is simply the one-hot encoding of the label.

We denote the true label of $\boldsymbol{x}$ as y^*. While noisy labels may arise in different ways, one common assumption is that, given the true labels, the noise is conditionally independent to the inputs, i.e., $\boldsymbol{q}(y = k|y^*= j, \boldsymbol{x}) = \boldsymbol{q}(y = k|y^* = j)$. Under this assumption, label noise can be either *symmetric* (or uniform), or *asymmetric* (or class-conditional). We denote the overall noise rate by $\eta \in [0, 1]$ and the class-wise noise rate from class j to class k by η_{jk}. Then, for symmetric noise, $\eta_{jk} = \frac{\eta}{K-1}$ for $j \neq k$ and $\eta_{jk} = 1 - \eta$ for $j = k$. For asymmetric noise, η_{jk} is conditioned on both the true class j and mislabeled class k.

Classification is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ (as represented by a DNN) that maps the input space to the label space. For a sample $\boldsymbol{x}$, we denote the probability output of a DNN classifier $f(\boldsymbol{x})$ as: $\boldsymbol{p}(k|\boldsymbol{x}) = \frac{e^{\boldsymbol{z}_k}}{\sum_{j=1}^{K} e^{\boldsymbol{z}_j}}$, where $\boldsymbol{z}_k$ denotes the logits output of the network with respect to class k. Training classifier f is to find a set of optimal parameters $\boldsymbol{\theta}$ that minimize the empirical risk defined by a loss function: $\boldsymbol{\theta} := \arg\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} \mathcal{L}(f(\boldsymbol{x}_i), y_i)$, where $\mathcal{L}(f(\boldsymbol{x}), y)$ is the loss of f with respect to label y. Next, we briefly introduce four loss functions that are either popularly used or recently proposed for robust classification with noisy labels.

Existing loss functions. The commonly used Cross Entropy (CE) loss on sample $\boldsymbol{x}$ is defined as: $CE = -\sum_{k=1}^{K} \boldsymbol{q}(k|\boldsymbol{x}) \log \boldsymbol{p}(k|\boldsymbol{x})$, which has been proved not robust to noisy labels (Ghosh et al., 2017).

Mean Absolute Error (MAE) is also a popular classification loss, and is defined as: $MAE = \sum_{k=1}^{K} |\boldsymbol{p}(k|\boldsymbol{x}) - \boldsymbol{q}(k|\boldsymbol{x})|$. MAE is provably robust to label noise (Ghosh et al., 2017).

The recently proposed Reverse Cross Entropy (RCE) loss (Wang et al., 2019c) is defined as: $RCE = -\sum_{k=1}^{K} \boldsymbol{p}(k|\boldsymbol{x}) \log \boldsymbol{q}(k|\boldsymbol{x})$, with $\boldsymbol{q}(k \neq y|\boldsymbol{x}) = 0$ is truncated to a small value such that $\log(\boldsymbol{q}(k \neq y|\boldsymbol{x})) = A$ (eg. $A = -4$). RCE has also been proved to be robust to label noise, and can be combined with CE to form the Symmetric Cross Entropy (SCE) for robust classification and boosted learning (Wang et al., 2019c).

Focal Loss (FL) (Lin et al., 2017), originally proposed for dense object detection, is also an effective loss function for classification. FL is also a generalization of the CE loss, and is defined as: $FL = -\sum_{k=1}^{K} \boldsymbol{q}(k|\boldsymbol{x})(1 - \boldsymbol{p}(k|\boldsymbol{x}))^\gamma \log \boldsymbol{p}(k|\boldsymbol{x})$, where $\gamma \geq 0$ is a tunable parameter. FL reduces to the CE loss when $\gamma = 0$, and is not robust to noisy labels following (Ghosh et al., 2017).

3.2. Normalized Loss Functions

Following (Ghosh et al., 2017; Charoenphakdee et al., 2019), we know that if a loss function $\mathcal{L}$ satisfies $\sum_{j}^{K} \mathcal{L}(f(\boldsymbol{x}), j) = C, \forall \boldsymbol{x} \in \mathcal{X}, \forall f$, where C is some constant, then $\mathcal{L}$ is noise tolerant under mild assumptions. Based on this, we propose to normalize a loss function by:

$$\mathcal{L}_{\text{norm}} = \frac{\mathcal{L}(f(\boldsymbol{x}), y)}{\sum_{j=1}^{K} \mathcal{L}(f(\boldsymbol{x}), j)}. \tag{1}$$

A normalized loss has the property: $\mathcal{L}_{\text{norm}} \in [0, 1]$.

Accordingly, we can normalize the above four loss functions defined in Section 3.1 as follows. The Normalized Cross Entropy (NCE) loss can be defined as:

$$NCE = \frac{-\sum_{k=1}^{K} \boldsymbol{q}(k|\boldsymbol{x}) \log \boldsymbol{p}(k|\boldsymbol{x})}{-\sum_{j=1}^{K} \sum_{k=1}^{K} \boldsymbol{q}(y = j|\boldsymbol{x}) \log \boldsymbol{p}(k|\boldsymbol{x})}$$
$$= \log_{\prod_{k}^{K} \boldsymbol{p}(k|\boldsymbol{x})} \boldsymbol{p}(y|\boldsymbol{x}), \tag{2}$$

where, the last equality holds following the change of base rule in logarithm (eg. $\log_a b = \frac{\log b}{\log a}$).

The Normalized Mean Absolute Error (NMAE) is:

$$NMAE = \frac{\sum_{k=1}^{K} |\boldsymbol{p}(k|\boldsymbol{x}) - \boldsymbol{q}(k|\boldsymbol{x})|}{\sum_{j=1}^{K} \sum_{k=1}^{K} |\boldsymbol{p}(k|\boldsymbol{x}) - \boldsymbol{q}(y = j|\boldsymbol{x})|}$$
$$= \frac{1}{K - 1}(1 - \boldsymbol{p}(y|\boldsymbol{x})) = \frac{1}{2(K - 1)} \cdot MAE. \tag{3}$$

The last two equalities hold due to $\sum_{k=1}^{K} |\boldsymbol{p}(k|\boldsymbol{x}) - \boldsymbol{q}(k|\boldsymbol{x})| = 2(1 - \boldsymbol{p}(y|\boldsymbol{x}))$. As can be observed, NMAE is simply a scaled version of MAE by a factor of $\frac{1}{2(K-1)}$.

The Normalized Reverse Cross Entropy (NRCE) loss is:

$$NRCE = \frac{-\sum_{k=1}^{K} \boldsymbol{p}(k|\boldsymbol{x}) \log \boldsymbol{q}(k|\boldsymbol{x})}{-\sum_{j=1}^{K} \sum_{k=1}^{K} \boldsymbol{p}(k|\boldsymbol{x}) \log \boldsymbol{q}(y = j|\boldsymbol{x})}$$
$$= \frac{1}{K - 1}(1 - \boldsymbol{p}(y|\boldsymbol{x})) = \frac{1}{A(K - 1)} \cdot RCE. \tag{4}$$

The last two equalies hold as $\sum_{k=1}^{K} \boldsymbol{p}(k|\boldsymbol{x}) \log \boldsymbol{q}(k|\boldsymbol{x}) = A(1 - \boldsymbol{p}(y|\boldsymbol{x}))$. Similar to NMAE, NRCE is a scaled version of RCE by a factor of $\frac{1}{A(K-1)}$.

The Normalized Focal Loss (NFL) can be defined as:

$$NFL = \frac{-\sum_{k=1}^{K} \boldsymbol{q}(k|\boldsymbol{x})(1 - \boldsymbol{p}(k|\boldsymbol{x}))^\gamma \log \boldsymbol{p}(k|\boldsymbol{x})}{-\sum_{j=1}^{K} \sum_{k=1}^{K} \boldsymbol{q}(y = j|\boldsymbol{x})(1 - \boldsymbol{p}(k|\boldsymbol{x}))^\gamma \log \boldsymbol{p}(k|\boldsymbol{x})}$$
$$= \log_{\prod_{k}^{K}(1-\boldsymbol{p}(k|\boldsymbol{x}))^\gamma \boldsymbol{p}(k|\boldsymbol{x})}(1 - \boldsymbol{p}(y|\boldsymbol{x}))^\gamma \boldsymbol{p}(y|\boldsymbol{x}). \tag{5}$$

Under this normalization scheme, the normalized forms of robust loss functions such as MAE and RCE are simply

a scaled version of their original forms. This keeps their robustness property. For the rest of this paper, we will use the original forms for MAE and RCE if not otherwise explicitly stated. On the contrary, normalization on non-robust loss functions such as CE and FL derives new loss functions. Note that the above four normalized losses are just a proof-of-concept, other loss functions can also be normalized following Eq. (1).

3.3. Theoretical Justification

Following previous works (Ghosh et al., 2017; Wang et al., 2019c), we can show that normalized loss functions are noise tolerant to both symmetric and asymmetric label noise.

Lemma 1. *In a multi-class classification problem, any normalized loss function $\mathcal{L}_{norm}$ is noise tolerant under symmetric (or uniform) label noise, if noise rate $\eta < \frac{K-1}{K}$.*

Lemma 2. *In a multi-class classification problem, given $R(f^*) = 0$ and $0 \leq \mathcal{L}_{norm}(f(\boldsymbol{x}), k) \leq \frac{1}{K-1}, \forall k$, any normalized loss function $\mathcal{L}_{norm}$ is noise tolerant under asymmetric (or class-conditional) label noise, if noise rate $\eta_{jk} < 1 - \eta_y$.*

Detailed proofs for Lemma 1 and Lemma 2 can be found in Appendix A. We denote the risk of classifier f under clean labels as $R(f) = \mathbb{E}_{\boldsymbol{x}, y^*} \mathcal{L}_{norm}$, and the risk under label noise rate η as $R^{\eta}(f) = \mathbb{E}_{\boldsymbol{x}, y} \mathcal{L}_{norm}$. Let f^* and f_{η}^* be the global minimizers of $R(f)$ and $R^{\eta}(f)$, respectively. We need to prove f^* is also a global minimizer of noisy risk $R^{\eta}(f)$ for $\mathcal{L}$ to be robust. The noise rate conditions in Lemma 1 ($\eta < \frac{K-1}{K}$) and Lemma 2 ($\eta_{jk} < 1 - \eta_y$) generally requires that the correct labels are still the majority of the class. In Lemma 2, the restrictive condition $R(f^*) = 0$ may not be satisfied in practice (eg. the classes may not completely separable), however, good empirical robustness can still be achieved. While the condition $0 \leq \mathcal{L}_{norm}(f(\boldsymbol{x}), k) \leq \frac{1}{K-1}, \forall k$ can be easily satisfied by a typical loss function. We refer the reader to (Charoenphakdee et al., 2019) for more discussions of other theoretical properties such as classification calibration.

So far, we have presented a somewhat surprising but theoretically justified result that any loss function can be made robust to noisy labels. This advances current theoretical progresses in this field. While this finding is exciting, in the following, we will empirically show that robustness alone is not sufficient for obtaining good performance.

4. Robustness Alone is not Sufficient

In this section, we empirically show that the above four robust loss functions (eg. NCE, NFL, MAE and RCE) all suffer from an underfitting problem, and thus are not sufficient by themselves to train accurate DNNs. We then propose a new framework to build loss functions that are both theoretically robust and learning sufficient.

Robust losses can suffer from underfitting. To motivate this problem, we use an example on CIFAR-100 dataset with 0.6 symmetric noise. We train a ResNet-34 (He et al., 2016) using both normalized and unnormalized loss functions (detailed setting can be found in Section 5.2). As can be observed in Figure 1, CE and FL losses become robust after normalization, however, this robustness does not lead to more accurate models. In fact, robust losses NCE and NFL demonstrate even worse performance than nonrobust CE and FL. Moreover, even without normalization, the originally robust loss functions MAE and RCE also suffer from underfitting: they even fail to converge in this scenario. We find that this underfitting issue occur across different training settings in terms of learning rate, learning rate scheduler, weight decay and the number of training epochs. We identify this problem as an *underfitting problem of existing robust loss functions*, at least for the four tested loss functions (eg. NCE, NFL, MAE and RCE). Next, we will propose a new loss framework to address this problem.
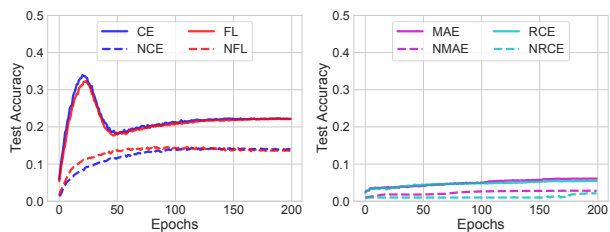

Figure 1. Test accuracies of unnormalized versus normalized loss functions on CIFAR-100 under 0.6 symmetric noise.

4.1. Proposed Active Passive Loss (APL)

In (Kim et al., 2019), the use of complementary labels ("input does not belong to this complementary class") together with the original labels was shown to help learning and robustness. In (Wang et al., 2019c), a Reverse Cross Entropy term was found can provide a robust boost to the CE loss. To generalize these works taking a loss function perspective, we characterize existing robust functions into two types: "Active" and "Passive", based on their optimization (maximization/minimization) behaviors.

At a high level, a loss is defined "Active" if it only optimizes at $\boldsymbol{q}(k = y|x) = 1$, otherwise, a loss is defined as "Passive". We denote the basic function of loss $\mathcal{L}(f(x), y)$ by $\ell(f(x), k)$, that is $\mathcal{L}(f(x), y) = \sum_{k=1}^{K} \ell(f(x), k)$. Then, we can define the active and passive loss functions as:

Definition 1. *(Active loss function)* $\mathcal{L}_{Active}$ *is an active loss function if* $\forall(\boldsymbol{x}, y) \in \mathcal{D} \ \forall k \neq y \ \ell(f(\boldsymbol{x}), k) = 0$.

Definition 2. *(Passive loss function)* $\mathcal{L}_{Passive}$ *is a passive loss function if* $\forall(\boldsymbol{x}, y) \in \mathcal{D} \ \exists k \neq y \ \ell(f(\boldsymbol{x}), k) \neq 0$.

According to the above two definitions, active losses only

explicitly maximize the network's output probability at the class position specified by the label y. For example in CE loss, only the probability at $q(k = y|x) = 1$ is explicitly maximized (the loss is zero at $q(k \neq y|x) = 0$). Different from active losses, passive losses also explicitly minimize the probability at at least one other class positions. For example in MAE, the probabilities at position $k \neq y$ are also explicitly minimized along with the maximization of the probability at $k = y$. Note that this characterization applies to both robust and nonrobust loss functions. Table 1 summarizes examples of active and passive losses.

Table 1. Examples of active and passive loss functions.

Loss Type	Active	Passive
Examples	CE, NCE, FL, NFL	MAE, NMAE, RCE, NRCE

Definition of APL. Inspired by the benefit of symmetric (Wang et al., 2019c) or complementary learning (Kim et al., 2019), we propose to combine a robust active loss and a robust passive loss into an "Active Passive Loss" (APL) framework for both robust and sufficient learning. Formally,

$$\mathcal{L}_{\text{APL}} = \alpha \cdot \mathcal{L}_{\text{Active}} + \beta \cdot \mathcal{L}_{\text{Passive}}, \qquad (6)$$

where, $\alpha, \beta > 0$ are parameters to balance the two terms. An important requirement for the two loss terms is robustness, which means a nonrobust loss should be normalized following Eq. (1) for it to be used within our APL scheme. This guarantees the robustness property of APL loss functions (proof can be found in Appendix A):

Lemma 3. $\forall \alpha, \forall \beta$, if $\mathcal{L}_{\text{Active}}$ and $\mathcal{L}_{\text{Passive}}$ are noise tolerant, then $\mathcal{L}_{\text{APL}} = \alpha \cdot \mathcal{L}_{\text{Active}} + \beta \cdot \mathcal{L}_{\text{Passive}}$ is noise tolerant.

In APL, the two loss terms optimize the same objective from two complementary directions (eg. maximizing $p(k = y|x)$ and minimizing $p(k \neq y|x)$). For the four loss functions considered in this paper, there are four possible combinations satisfying our APL principle: 1) $\alpha NCE + \beta MAE$, 2) $\alpha NCE + \beta RCE$, 3) $\alpha NFL + \beta MAE$ and 4) $\alpha NFL + \beta RCE$. For simplicity we omit the parameters α, β in the rest of this paper. According to our active/passive definitions, APL losses can be considered as passive losses. However, APL losses are different from passive losses that have only one term, since they contain at least two terms and one of them is an active loss term. Whilst different choices of the two loss terms may lead to different performance, we will show in Section 5 that APL losses generally achieve better or at least comparable performance to state-of-the-art noisy label learning methods.

4.2. More Insights into APL Loss Functions

Here, we provide some insights into the underfitting issue of robust loss functions, and why the proposed APL losses can address underfitting.

Why robust loss functions underfit? Taking the NCE loss defined in Eq. (2) as an example, the underfitting is caused by the extra terms introduced into the denominator by the normalization. In Eq. (2), NCE is in the form of $\frac{P}{P+Q}$, where $P = -\log(p_y)$ and $Q = -\sum_{k \neq y} \log(p_k)$. During training, the Q term may increase even when P is fixed (eg. p_y is fixed), and it reaches the highest value when all $p_{k \neq y}$ equals to $(1 - p_y)/(K - 1)$ (eg. the highest entropy). This implies that the network may learn nothing for the prediction (as p_y is fixed) even when the loss decreases (as Q increases). This tends to hinder the convergence and cause the underfitting problem. Other robust loss functions such as MAE and RCE all suffer from a similar issue.

Why APL can address underfitting? APL combines an active loss with a passive loss. By definition, the passive loss explicitly minimizes (at least one component of) the Q term discussed above so that it won't increase when p_y is fixed. This directly addresses the underfitting issue of a robust active loss. Therefore, APL losses can leverage both the robustness and the convergence advantages. Note that, by definition, passive loss has a broader scope than active loss. A single passive loss like MAE can be decomposed into an active term and a passive term, with the two terms already form an APL loss. With proper balancing between the two terms, the reformulated MAE can also be a powerful new loss. For example, a recent work has shown that a reweighted MAE can outperform CE (Wang et al., 2019a).

4.3. Connection to Related Work

Our APL framework is a generalization of several state-of-the-art methods. Following APL, better performance can be achieved with existing loss functions, rather than complex modifications on the training procedure. Although NLNL (Kim et al., 2019) can improve robustness with complementary labels, it has slow convergence ($10\times$ slower than standard training), and requires a complex 3-stage training procedure: 1) training with complementary labels, 2) training with high confidence (above a threshold) complementary labels, and 3) training with high confidence original labels. From our APL perspective, NLNL switches back and forth between active learning (with original labels) and passive learning (with complementary labels). Such a learning scheme can instead be achieved alternatively using our APL. Indeed, when defined on complementary labels, the CE loss becomes -1/(C-1)log(1-RCE) with A=-1 in RCE, and our APL loss NCE+RCE can be seen as a simpler alternative for NLNL. Compared to the SCE (Wang et al., 2019c) loss (eg. CE+RCE), our APL loss NCE+RCE can be seen as its normalized version, which has theoretically guaranteed robustness. This modification to SCE can improve its performance considerably (see Section 5.2). Compared to the GCE loss (Zhang & Sabuncu, 2018) which can be regraded as a mixture of CE and MAE, our APL loss NCE+MAE

is an alternative solution that directly adds the two terms together with normalization. NCE+MAE is theoretically robust while GCE is not. Moreover, the GCE loss itself can be normalized and improved following our APL framework (see Section 5.3).

5. Experiments

In this section, we empirically investigate our proposed APL loss functions on benchmark datasets MNIST (LeCun et al., 1998), CIFAR-10/-100 (Krizhevsky & Hinton, 2009), and a real-world noisy dataset WebVision (Li et al., 2017a).

5.1. Empirical Understandings

Normalized losses are robust. We first run a set of experiments on CIFAR-10 and CIFAR-100 to verify whether non-robust losses CE and FL become robust after normalization (NCE and NFL). We set the label noise to be symmetric, and the noise rate to 0.6 for both CIFAR-10 and CIFAR-100. We use an 8-layer convolutional neural network (CNN) for CIFAR-10 and a ResNet-34 (He et al., 2016) for CIFAR-100. On each dataset, we train the same network using different loss functions, eg. normalized versus unnormalized. For FL/NFL loss we set $\gamma = 0.5$, while for RCE/NRCE loss, we set $A = -4$. Detailed settings are in Section 5.2.

As shown in Figures 1 & 2, both CE and FL losses exhibit significant overfitting after epoch 25. However, as we have theoretically proved, their normalized forms (eg. NCE and NFL) are robust: no overfitting was observed during the entire training process. Moreover, for the already robust loss functions MAE and RCE, normalization does not break their robustness property. We observe the same results across different datasets (eg. MNIST, CIFAR-10 and CIFAR-100) under different noise rates ($\eta \in [0.2, 0.8]$). In general, the higher the noise rate, the more overfitting of nonrobust loss functions, and their normalized forms are always robust. This empirically verifies our theoretical finding that any loss can be made robust following normalization in Eq. (1).

Can scaling help sufficient learning? As one may have noticed in Figures 1 & 2, NMAE and NRCE exhibit more severe underfitting than MAE and RCE, even though they are just scaled versions of MAE and RCE. This raises the question: can the underfitting problem be addressed by scaling the normalized losses up by a factor? In Figure 3, we show different scales applied to NCE, NFL, MAE and RCE for training on CIFAR-100 with 0.6 symmetric noise. We find that scaled NCE and NFL only slightly improve learning after epoch 150, when the learning rate is decayed to be smaller. This is because scaling the loss is equivalent to scaling the gradients, a similar effect to increasing the learning rate. Moreover, scaled MAE and RCE still fail to converge in this scenario. This highlights
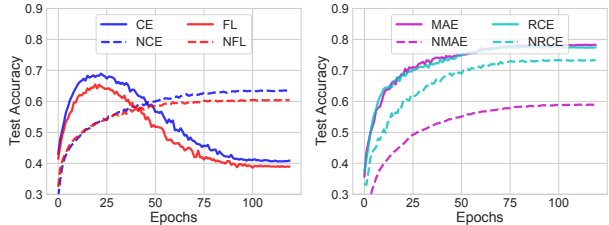

Figure 2. Test accuracies of unnormalized versus normalized loss functions on CIFAR-10 under 0.6 symmetric noise.

(a) Scaled NCE   (b) Scaled NFL
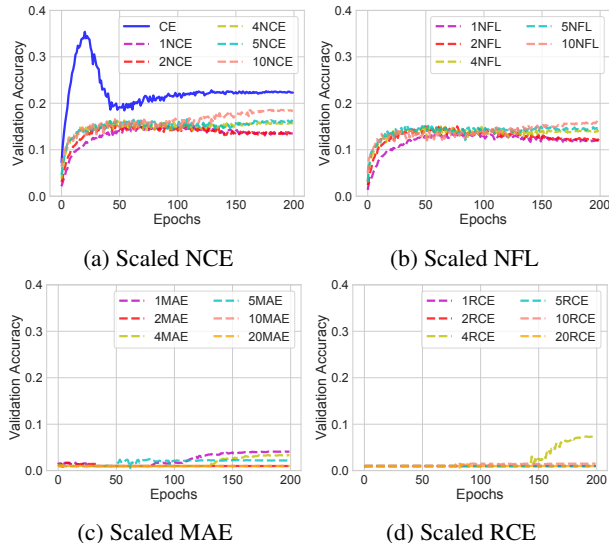
(c) Scaled MAE   (d) Scaled RCE

Figure 3. Test accuracies of scaled loss functions on CIFAR-100 with 0.6 symmetric noise.

that scaling may not be an effective solution for sufficient learning, especially for challenging datasets like CIFAR-100. On the simple dataset CIFAR-10, proper scaling does help learning. But this can alternatively can be achieved by adjusting the learning rate.

APL losses are both robust and learning sufficient. We show the effectiveness of "Active+Passive" learning, compared to other forms of combinations. We run experiments on CIFAR-10 and CIFAR-100 under the same settings as above. The parameters α, β for our APL are simply set to 1.0 without any tuning. As shown in Figure 4, the 4 APL loss functions demonstrate a clear advantage over either AAL ("Active+Active Loss") or PPL ("Passive+Passive Loss"), especially for sufficient learning (high accuracy). The AAL and PPL loss functions are robust but still suffer from the underfitting problem. This highlights that the overfitting and underfitting problems can be addressed simultaneously by the joint of active and passive losses by our APL.

Parameter Analysis of APL. We tune the parameters α and β for NCE+RCE loss, then directly use these parameters for all other APL losses. This is also done on CIFAR-10 and CIFAR-100 datasets under 0.6 symmetric noise. We test the combinations between $\alpha \in \{0.1, 1.0, 10.0\}$ and $\beta \in$
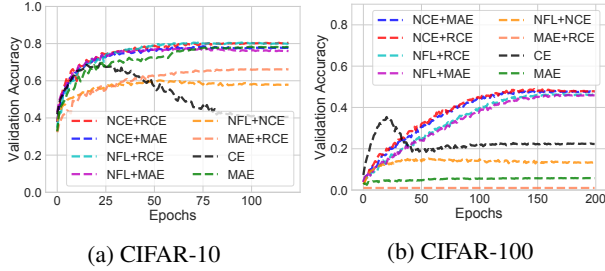
(a) CIFAR-10   (b) CIFAR-100

Figure 4. Test accuracies of APL loss functions (NCE+MAE, NCE+RCE, NFL+MAE and NFL+RCE) versus "AAL" loss (NFL+NCE) or "PPL" loss (MAE+RCE) on CIFAR-10/CIFAR-100 with 0.6 symmetric noise.

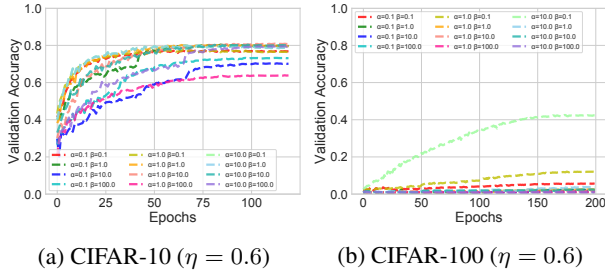(a) CIFAR-10 ($\eta = 0.6$)   (b) CIFAR-100 ($\eta = 0.6$)

Figure 5. Validation accuracy of NCE+RCE loss with different parameters on CIFAR-10 and CIFAR-100 under symmetric noise.

$\{0.1, 1.0, 10.0, 100.0\}$, then select the optimal combination according to the validation accuracy on a randomly sampled validation set (20% training data). As shown in Figure 5, the optimal parameters for CIFAR-10 are $\alpha = 1, \beta = 1$, and CIFAR-100 are $\alpha = 10, \beta = 0.1$. In general, on more complex dataset (eg. CIFAR-100 > CIFAR-10), it requires more active learning (eg. a large α) and less passive learning (eg. a small β) to achieve good performance.

5.2. Evaluation on Benchmark Datasets

Baselines. We consider 3 state-of-the-art methods: 1) Generalized Cross Entropy (GCE) (Zhang & Sabuncu, 2018); 2) Negative Learning for Noisy Labels (NLNL) (Kim et al., 2019); and 3) Symmetric Cross Entropy (SCE) (Wang et al., 2019c). For APL, we consider 4 loss functions: 1) NCE+MAE, 2) NCE+RCE, 3) NFL+MAE and 4) NFL+RCE. We also train networks using CE and FL losses.

Noise generation. The noisy labels are generated following standard approaches in previous works (Patrini et al., 2017; Ma et al., 2018). Symmetric noise is generated by flipping labels in each class randomly to incorrect labels of other classes. For asymmetric noise, we flip the labels within a specific set of classes. For CIFAR-10, flipping TRUCK $\rightarrow$ AUTOMOBILE, BIRD $\rightarrow$ AIRPLANE, DEER $\rightarrow$ HORSE, CAT $\leftrightarrow$ DOG. For CIFAR-100, the 100 classes are grouped into 20 super-classes with each has 5 sub-classes, we then flip each class within the same super-class into the next in a circular fashion. We vary the noise rate $\eta \in [0.2, 0.8]$ for

symmetric noise, and $\eta \in [0.1, 0.4]$ for asymmetric noise.

Networks and training. We use a 4-layer CNN for MNIST, an 8-layer CNN for CIFAR-10 and a ResNet-34 for CIFAR-100. We train the networks for 50, 120 and 200 epochs for MNIST, CIFAR-10, and CIFAR-100, respectively. For all the training, we use SGD optimizer with momentum 0.9 and cosine learning rate annealing. Weight decay is set to $1 \times 10^{-3}, 1 \times 10^{-4}$ and 1×10^{-5} for MNIST, CIFAR-10 and CIFAR-100, respectively. The initial learning rate is set to 0.01 for MNIST/CIFAR-10 and 0.1 for CIFAR-100. Typical data augmentations including random width/height shift and horizontal flip are applied.

Parameter setting. We tune the parameters for all baseline methods and find that the optimal settings match their original papers. Specifically, for GCE, we set $\rho = 0.7$ (see detailed definition in Section 5.3). For SCE, we set $A = -4$, and $\alpha = 0.01, \beta = 1.0$ for MNIST, $\alpha = 0.1, \beta = 1.0$ for CIFAR-10, $\alpha = 6.0, \beta = 0.1$ for CIFAR-100. For FL, we set $\gamma = 0.5$. For our APL losses, we empirically set $\alpha = 1, \beta = 100$ for MNIST, $\alpha, \beta = 1$ for CIFAR-10, and $\alpha = 10, \beta = 0.1$ for CIFAR-100.

Results. The classification accuracies under symmetric label noise are reported in Table 2. As can be seen, our APL loss functions achieved the top 2 best results in all test scenarios across all datasets. The superior performance of APL losses is more pronounced when the noise rates are extremely high and the dataset is more complex. For example, on CIFAR-10 with 0.6 symmetric noise, our APL losses NFL+RCE and NCE+RCE outperform the state-of-the-art robustness (72.85% of NLNL) by more than 6%. On CIFAR-100 with 0.8 symmetric noise where CE and FL both fail to converge, our NCE+MAE and NCE+RCE outperform the state-of-the-art methods GCE and NLNL by at least 9%. In several cases, all our 4 APL losses are better than baseline methods.

Results for asymmetric noise are reported in Table 3. Again, all top 2 best results are achieved by our APL loss functions across different datasets and noise rates. On CIFAR-100 with 0.4 asymmetric noise, the highest accuracy that can be achieved by current methods is 42.19% (by SCE), which is still 5% lower than our NCE+MAE and 4% lower than our NCE+RCE. Comparing results in both Table 2 and Table 3, we find that the best combination of our APL loss varies across different datasets, but within the same dataset, is quite consistent across different noise types and noise rates.

Overall, NCE+RCE demonstrates a consistently strong performance across different datasets. The strong performance of our APL losses verifies the importance of theoretically guaranteed robustness and "Active+Passive" learning. Our proposed APL framework can be used as a general principle for developing new robust loss functions.

Table 2. Test accuracies (%) of different methods on benchmark datasets with clean or symmetric label noise ($\eta \in [0.2, 0.8]$). The results (mean±std) are reported over 3 random runs and the top 2 best results are **boldfaced**.

Datasets	Methods	Clean (η=0.0)	Symmetric Noise Rate (η)			
			0.2	0.4	0.6	0.8
MNIST	CE	99.25 ± 0.08	97.42 ± 0.06	94.21 ± 0.54	86.00 ± 1.48	47.08 ± 1.15
	FL	99.30 ± 0.02	97.45 ± 0.19	94.71 ± 0.25	85.76 ± 1.85	49.77 ± 2.26
	GCE	99.27 ± 0.01	99.18 ± 0.06	98.72 ± 0.05	97.43 ± 0.23	12.77 ± 2.00
	NLNL	99.27 ± 0.02	97.49 ± 0.30	96.64 ± 0.52	97.22 ± 0.06	10.32 ± 0.73
	SCE	99.24 ± 0.08	99.15 ± 0.04	98.78 ± 0.09	97.45 ± 0.29	73.70 ± 0.84
	NFL+MAE	99.39 ± 0.04	99.12 ± 0.06	98.74 ± 0.14	96.91 ± 0.09	$\mathbf{74.98 \pm 1.99}$
	NFL+RCE	99.38 ± 0.02	$\mathbf{99.19 \pm 0.06}$	$\mathbf{98.79 \pm 0.10}$	$\mathbf{97.46 \pm 0.03}$	74.59 ± 2.23
	NCE+MAE	99.37 ± 0.02	99.14 ± 0.05	98.78 ± 0.00	96.76 ± 0.34	74.66 ± 1.11
	NCE+RCE	99.37 ± 0.02	$\mathbf{99.20 \pm 0.04}$	$\mathbf{98.79 \pm 0.12}$	$\mathbf{97.48 \pm 0.13}$	$\mathbf{75.18 \pm 1.19}$
CIFAR-10	CE	90.36 ± 0.03	75.90 ± 0.28	60.28 ± 0.27	40.90 ± 0.35	19.65 ± 0.46
	FL	89.63 ± 0.25	74.59 ± 0.49	57.55 ± 0.39	38.91 ± 0.62	19.43 ± 0.27
	GCE	89.38 ± 0.23	87.27 ± 0.21	83.33 ± 0.39	72.00 ± 0.37	29.08 ± 0.80
	NLNL	91.93 ± 0.20	83.98 ± 0.18	76.58 ± 0.44	72.85 ± 0.39	51.41 ± 0.85
	SCE	91.30 ± 0.22	88.05 ± 0.26	82.06 ± 0.24	66.08 ± 0.25	30.69 ± 0.63
	NFL+MAE	89.25 ± 0.19	87.33 ± 0.14	83.81 ± 0.06	76.36 ± 0.31	45.23 ± 0.52
	NFL+RCE	90.91 ± 0.02	$\mathbf{89.14 \pm 0.13}$	$\mathbf{86.05 \pm 0.12}$	$\mathbf{79.78 \pm 0.13}$	$\mathbf{55.06 \pm 1.08}$
	NCE+MAE	88.83 ± 0.34	87.12 ± 0.21	84.19 ± 0.43	77.61 ± 0.05	49.62 ± 0.72
	NCE+RCE	90.76 ± 0.22	$\mathbf{89.22 \pm 0.27}$	$\mathbf{86.02 \pm 0.09}$	$\mathbf{79.78 \pm 0.50}$	52.71 ± 1.90
CIFAR-100	CE	70.89 ± 0.22	56.99 ± 0.41	41.40 ± 0.36	22.15 ± 0.40	7.58 ± 0.44
	FL	70.61 ± 0.44	56.10 ± 0.48	40.77 ± 0.62	22.14 ± 1.00	7.21 ± 0.25
	GCE	69.00 ± 0.56	65.24 ± 0.56	58.94 ± 0.50	45.18 ± 0.93	16.18 ± 0.46
	NLNL	68.72 ± 0.60	46.99 ± 0.91	30.29 ± 1.64	16.60 ± 0.90	11.01 ± 2.48
	SCE	70.38 ± 0.45	55.39 ± 0.18	39.99 ± 0.59	22.35 ± 0.65	7.57 ± 0.28
	NFL+MAE	67.98 ± 0.52	63.58 ± 0.09	58.18 ± 0.08	46.10 ± 0.50	24.78 ± 0.82
	NFL+RCE	68.23 ± 0.62	64.52 ± 0.35	58.20 ± 0.31	46.30 ± 0.45	25.16 ± 0.55
	NCE+MAE	68.75 ± 0.54	$\mathbf{65.25 \pm 0.62}$	$\mathbf{59.22 \pm 0.36}$	$\mathbf{48.06 \pm 0.34}$	25.50 ± 0.76
	NCE+RCE	69.02 ± 0.11	$\mathbf{65.31 \pm 0.07}$	$\mathbf{59.48 \pm 0.56}$	47.12 ± 0.62	$\mathbf{25.80 \pm 1.12}$

5.3. Improving New Loss Functions using APL

Next, we take GCE (Zhang & Sabuncu, 2018) as an example and show how to improve a new loss function using our APL framework. Given a sample x, GCE loss is defined as: $GCE = \sum_{k=1}^{K} q(k|x) \frac{1-p(k|x)^\rho}{\rho}$, where $\rho \in (0, 1]$. GCE reduces to the MAE/unhinged loss and CE loss when $\rho = 1$ and $\rho \to 0$, respectively. Following Eq. (1), the Normalized Generalized Cross Entropy (NGCE) loss can be defined as: $NGCE = (1 - p(y|x)^\rho)/(K - \sum_{k=1}^{K} p(k|x)^\rho)$.

Both GCE and NGCE are active loss functions (eg. $\ell(f(x), k) = 0, \forall k \neq y$). Thus, following our APL in Eq. (6), we can define two APL losses for NGCE: 1) NGCE+MAE and 2) NGCE+RCE. Here, we simply set $\alpha, \beta = 1.0$ for both APL losses. We compare their performance to GCE (with $\rho = 0.7$) on CIFAR-10 under both symmetric and asymmetric noise. As shown in Table 4, both NGCE+MAE and NGCE+RCE can improve the performance of GCE under different noise settings, except for NGCE+RCE under 0.4 asymmetric noise. Particularly, under 0.8 symmetric noise, NGCE+MAE is able to improve GCE by > 20%. A new loss function may have multiple terms, in this case, we can normalize its non-robust terms, and then add an active or passive loss into the loss function if there are missing.

5.4. Effectiveness on Real-world Noisy Labels

Here, we test the effectiveness of our APL loss functions on large-scale real-world noisy dataset WebVision (Li et al., 2017a). WebVision contains 2.4 million images of real-world noisy labels, crawled from the web (eg. Flickr and Google) based on the 1,000 class labels of ImageNet ILSVRC12 (Deng et al., 2009). Here, we follow the "Mini" setting in (Jiang et al., 2018) that only takes the first 50 classes of the Google resized image subset. We evaluate the trained networks on the same 50 classes of the ILSVRC12 validation set, which can be considered as a clean validation. We compare our APL losses NCE+MAE and NCE+RCE with GCE and SCE. For each loss, we train a ResNet-50 (He et al., 2016) using SGD for 250 epochs with initial learning rate 0.4, nesterov momentum 0.9 and weight decay 3×10^{-5} and batch size 512. The learning rate is multiplied by 0.97 after every epoch of training. We resize the images to 224×224. Typical data augmentations including random width/height shift, color jittering and random horizontal flip are applied. For GCE, we use the suggested $\alpha = 0.7$, while for SCE, we use the setting with $A = -4, \alpha = 10.0, \beta = 1.0$. For our two APL losses, we set $\alpha = 50.0, \beta = 0.1$ for NCE+RCE and $\alpha = 50.0, \beta = 1.0$ for NCE+MAE. The top-1 validation ac-

Table 3. Test accuracy (%) of different methods on benchmark datasets with clean or asymmetric label noise ($\eta \in [0.1, 0.4]$). The results (mean±std) are reported over 3 random runs and the top 2 best results are **boldfaced**.

Datasets	Methods	Asymmetric Noise Rate (η)			
		0.1	0.2	0.3	0.4
MNIST	CE	98.53 ± 0.11	96.75 ± 0.31	92.98 ± 1.41	85.74 ± 2.70
	FL	98.97 ± 0.10	98.35 ± 0.17	96.57 ± 0.36	91.18 ± 2.02
	GCE	99.25 ± 0.03	99.11 ± 0.04	96.99 ± 0.53	88.56 ± 2.40
	NLNL	98.38 ± 0.17	95.98 ± 0.58	91.52 ± 1.14	86.36 ± 0.40
	SCE	99.15 ± 0.07	99.05 ± 0.05	97.96 ± 0.40	91.89 ± 3.32
	NFL+MAE	99.31 ± 0.05	99.09 ± 0.12	97.88 ± 0.16	$\mathbf{93.52 \pm 0.19}$
	NFL+RCE	$\mathbf{99.33 \pm 0.06}$	99.13 ± 0.01	$\mathbf{97.99 \pm 0.05}$	$\mathbf{93.59 \pm 0.82}$
	NCE+MAE	99.26 ± 0.02	$\mathbf{99.21 \pm 0.04}$	$\mathbf{98.99 \pm 0.03}$	93.40 ± 1.28
	NCE+RCE	$\mathbf{99.34 \pm 0.06}$	99.17 ± 0.02	97.94 ± 0.21	93.12 ± 1.17
CIFAR-10	CE	87.38 ± 0.16	83.62 ± 0.15	79.38 ± 0.28	75.00 ± 0.50
	FL	86.35 ± 0.30	82.97 ± 0.14	79.48 ± 0.21	74.60 ± 0.15
	GCE	88.42 ± 0.07	86.07 ± 0.31	80.78 ± 0.21	74.98 ± 0.32
	NLNL	88.54 ± 0.25	84.74 ± 0.08	81.26 ± 0.43	76.97 ± 0.52
	SCE	88.13 ± 0.21	83.92 ± 0.07	79.70 ± 0.27	78.20 ± 0.03
	NFL+MAE	88.46 ± 0.20	86.81 ± 0.32	83.91 ± 0.34	77.16 ± 0.10
	NFL+RCE	$\mathbf{90.20 \pm 0.15}$	$\mathbf{88.73 \pm 0.29}$	$\mathbf{85.74 \pm 0.22}$	79.27 ± 0.43
	NCE+MAE	88.25 ± 0.09	86.44 ± 0.23	83.98 ± 0.52	78.23 ± 0.42
	NCE+RCE	$\mathbf{89.95 \pm 0.20}$	$\mathbf{88.56 \pm 0.17}$	$\mathbf{85.58 \pm 0.44}$	$\mathbf{79.59 \pm 0.40}$
CIFAR-100	CE	65.42 ± 0.22	58.45 ± 0.45	51.09 ± 0.29	41.68 ± 0.45
	FL	64.79 ± 0.18	58.59 ± 0.81	51.26 ± 0.18	42.15 ± 0.44
	GCE	61.98 ± 0.81	59.99 ± 0.83	53.99 ± 0.29	41.49 ± 0.79
	NLNL	59.55 ± 1.22	50.19 ± 0.56	42.81 ± 1.13	35.10 ± 0.20
	SCE	64.15 ± 0.61	58.22 ± 0.47	49.85 ± 0.91	42.19 ± 0.19
	NFL+MAE	$\mathbf{66.06 \pm 0.23}$	$\mathbf{63.10 \pm 0.22}$	56.19 ± 0.61	43.51 ± 0.42
	NFL+RCE	$\mathbf{66.13 \pm 0.31}$	$\mathbf{63.12 \pm 0.41}$	54.72 ± 0.38	42.97 ± 1.03
	NCE+MAE	65.71 ± 0.34	62.38 ± 0.60	$\mathbf{58.02 \pm 0.48}$	$\mathbf{47.22 \pm 0.30}$
	NCE+RCE	65.68 ± 0.25	62.68 ± 0.79	$\mathbf{57.82 \pm 0.41}$	$\mathbf{46.79 \pm 0.96}$

Table 4. Test accuracy (%) of APL losses NGCE+MAE and NGCE+RCE on CIFAR-10 under both symmetric and asymmetric noise. The top-2 best results are in **bold**.

Methods	Symmetric noise		Asymmetric noise
	0.4	0.8	0.4
GCE	83.33 ± 0.39	29.08 ± 0.80	$\mathbf{74.98 \pm 0.32}$
NGCE+MAE	$\mathbf{84.14 \pm 0.15}$	$\mathbf{50.55 \pm 1.08}$	$\mathbf{76.55 \pm 0.48}$
NGCE+RCE	$\mathbf{85.76 \pm 0.26}$	44.69 ± 4.93	71.65 ± 0.68

Table 5. Top-1 validation accuracies (%) on clean ILSVRC12 validation set of ResNet-50 models trained on WebVision using different loss functions, under the Mini setting in (Jiang et al., 2018). The top-2 best results are in **bold**.

Loss	CE	GCE	SCE	**NCE+MAE**	**NCE+RCE**
Acc	58.88	53.68	61.76	**62.36**	**62.64**

curacies of different loss functions on the clean ILSVRC12 validation set (eg. only the first 50 classes) are reported in Table 5. As can be observed, both our APL losses outperform existing loss functions GCE and SCE by a clear margin. This verifies the effectiveness of our APL against real-world label noise.

6. Conclusions

In this paper, we revisited the robustness and sufficient learning properties of existing loss functions for deep learning with noisy labels. We revealed a new theoretical insight into robust loss functions that: *a simple normalization can make any loss function robust to noisy labels.* Then, we highlighted that robustness alone is not enough for a loss function to train accurate DNNs, and existing robust loss functions all suffer from an underfitting problem. To address this problem, we characterize existing robust loss functions into "Active" or "Passive" losses, and then proposed a mutually boosted framework *Active Passive Loss* (APL). APL allows us to create a family of new loss functions that not only have theoretically guaranteed robustness but also are effective for sufficient learning. We empirically verified the excellent performance of our APL loss functions compared to state-of-the-art methods on benchmark datasets. Our APL framework can serve as a basic principle for developing new robust loss functions.

References

Charoenphakdee, N., Lee, J., and Sugiyama, M. On symmetric losses for learning from corrupted labels. In *ICML*, pp. 961–970, 2019.

Deng, J., Dong, W., Socher, R., Li, Li, J., Li, K., and Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Ghosh, A., Kumar, H., and Sastry, P. Robust loss functions under label noise for deep neural networks. In *AAAI*, 2017.

Goldberger, J. and Ben-Reuven, E. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017.

Han, B., Yao, J., Niu, G., Zhou, M., Tsang, I., Zhang, Y., and Sugiyama, M. Masking: A new perspective of noisy supervision. In *NeurIPS*, 2018a.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: robust training deep neural networks with extremely noisy labels. In *NeurIPS*, 2018b.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

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

Kim, Y., Yim, J., Yun, J., and Kim, J. Nlnl: Negative learning for noisy labels. In *CVPR*, pp. 101–110, 2019.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

Kumar, A. and Ithapu, V. K. Secost: Sequential co-supervision for weakly labeled audio event detection. *arXiv preprint arXiv:1910.11789*, 2019.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, K.-H., He, X., Zhang, L., and Yang, L. Cleannet: Transfer learning for scalable image classifier training with label noise. *arXiv preprint arXiv:1711.07131*, 2017.

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

Li, Y., Yang, J., Song, Y., Cao, L., Luo, J., and Li, J. Learning from noisy labels with distillation. In *ICCV*, 2017b.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *ICCV*, pp. 2980–2988, 2017.

Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S. M., Xia, S.-T., Wijewickrema, S., and Bailey, J. Dimensionality-driven learning with noisy labels. In *ICML*, 2018.

Malach, E. and Shalev-Shwartz, S. Decoupling" when to update" from" how to update". In *NeurIPS*, 2017.

Natarajan, N., Dhillon, I. S., Ravikumar, P. K., and Tewari, A. Learning with noisy labels. In *NeurIPS*, pp. 1196–1204, 2013.

Patrini, G., Rozza, A., Menon, A., Nock, R., and Qu, L. Making neural networks robust to label noise: a loss correction approach. In *CVPR*, 2017.

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., and Hinton, G. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

Reed, S., Lee, H., Anguelov, D., Szegedy, C., Erhan, D., and Rabinovich, A. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Sukhbaatar, S., Bruna, J., Paluri, M., Bourdev, L., and Fergus, R. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Tanaka, D., Ikami, D., Yamasaki, T., and Aizawa, K. Joint optimization framework for learning with noisy labels. In *CVPR*, 2018.

Vahdat, A. Toward robustness against label noise in training deep discriminative neural networks. In *NeurIPS*, 2017.

Veit, A., Alldrin, N., Chechik, G., Krasin, I., Gupta, A., and Belongie, S. Learning from noisy large-scale datasets with minimal supervision. In *CVPR*, 2017.

Wang, X., Hua, Y., Kodirov, E., and Robertson, N. M. Imae for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters. *arXiv preprint arXiv:1903.12141*, 2019a.

Wang, X., Kodirov, E., Hua, Y., and Robertson, N. M. Derivative manipulation for adjusting emphasis density function: A general example weighting framework. *arXiv preprint arXiv:1905.11233*, 2019b.

Wang, Y., Liu, W., Ma, X., Bailey, J., Zha, H., Song, L., and Xia, S.-T. Iterative learning with open-set noisy labels. In *CVPR*, 2018.

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. Symmetric cross entropy for robust learning with noisy labels. In *ICCV*, pp. 322–330, 2019c.

Xiao, T., Xia, T., Yang, Y., Huang, C., and Wang, X. Learning from massive noisy labeled data for image classification. In *CVPR*, 2015.

Xu, Y., Cao, P., Kong, Y., and Wang, Y. L_dmi: An information-theoretic noise-robust loss function. In *NeurIPS*, 2019.

Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., and Sugiyama, M. How does disagreement help generalization against label corruption? In *ICML*, 2019.

Zhang, Z. and Sabuncu, M. R. Generalized cross entropy loss for training deep neural networks with noisy labels. In *NeurIPS*, 2018.