DYNAMIC LOSS FOR LEARNING WITH LABEL NOISE

Anonymous authors
Paper under double-blind review

ABSTRACT

Label noise is verified seriously harmful to deep neural networks (DNNs). A simple and scalable strategy to handle this problem is to design robust loss functions, which improve generalization in the presence of label noise by reconciling fitting ability with robustness. However, the widely-used static trade-off between the two contradicts the dynamics of DNNs learning with label noise, leading to an inferior performance. Therefore, in this paper, we propose a dynamic loss function to solve this problem. Specifically, DNNs tend to first learn generalized patterns, then gradually overfit label noise. In light of this, we make fitting ability stronger initially, then gradually increase the weight of robustness. Moreover, we let DNNs put more emphasis on easy examples than hard ones at the later stage since the former are correctly labeled with a higher probability, further reducing the negative impact of label noise. Extensive experimental results on various benchmark datasets demonstrate the state-of-the-art performance of our method. We will open-source our code very soon.

1 INTRODUCTION

Deep neural networks (DNNs) have achieved tremendous success in a variety of tasks, particularly in supervised learning. However, their superior performance relies heavily on correctly annotated large-scale datasets. Considering that it is pretty expensive and time-consuming to manually label a big dataset without any error, one may prefer some cheap but imperfect methods such as querying search engines to collect labeled data, which inevitably introduce label noise. Unfortunately, DNNs can easily fit an entire training dataset with any ratio of noisy labels (Zhang et al., 2021a), which eventually results in poor generalization performance. Therefore, developing robust algorithms against label noise for DNNs is of great practical importance.

A simple and scalable way of handling label noise is to devise robust loss functions (Zhang & Sabuncu, 2018; Wang et al., 2019b; Amid et al., 2019; Ma et al., 2020; Feng et al., 2021; Zhou et al., 2021b;c). They typically cause no changes to the training process, require no extra information such as the noise rate or a clean validation set, and incur no additional memory burden or computational cost. It is widely observed that cross entropy (CE) often leads to serious overfitting in the presence of label noise due to its strong fitting ability. Meanwhile, although Ghosh et al. (2017) theoretically proved that mean absolute error (MAE) is robust against label noise, it suffers from severe underfitting in practice, especially on complex datasets. In light of this, many robust loss functions have been proposed to improve generalization by reconciling fitting ability with robustness, among which the generalized cross entropy (GCE) (Zhang & Sabuncu, 2018) is the most representative method. It is an interpolation between CE and MAE with a hyper-parameter $q \in (0, 1)$. As shown in Figure 1, selecting a suitable value for q such as 0.7, GCE ourperforms both CE and MAE significantly.

Besides devising robust learning algorithms, some recent work focused on delving deep into the dynamics of DNNs learning with label noise. Arpit et al. (2017) observed that while DNNs are capable of memorizing noisy labels perfectly, there are noticeable differences in DNNs' learning status at different time steps of the training process. Specifically, DNNs tend to first learn generalized patterns shared by the majority of training examples, then gradually overfit label noise. Further evidence is provided by Ma et al. (2018) that DNNs first learn simple representations via subspace dimensionality compression, then memorize noisy labels through subspace dimensionality expansion. This phenomenon is also verified by Figure 1 that the training accuracy on correctly label data always increases earlier than that on wrongly labeled data.

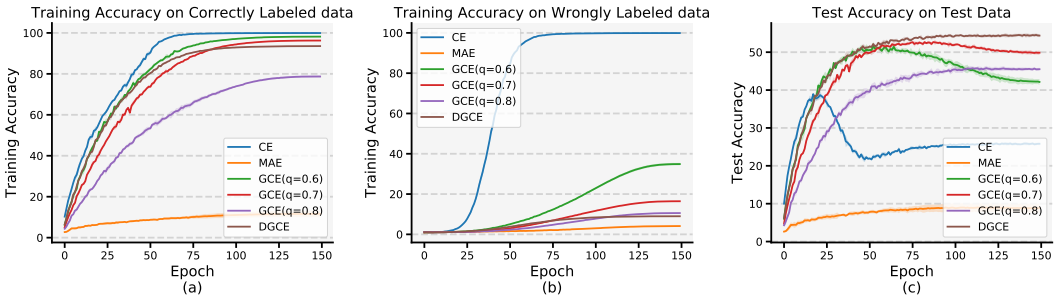

Figure 1: Performance of ResNet18 on CIFAR-100 with 60% symmetric noise, where the training accuracy calculation is based on the noisy labels. GCE with $q = 0.7$ achieves the highest test accuracy, outperforming both CE and MAE significantly, but the figure still declines remarkably at the later stage of the training process. Increasing q to 0.8, the training accuracy on wrongly labeled data remains low and the test accuracy grows steadily throughout the training process, but the training accuracy on correctly labeled data also remains at a low level, leading to a worse generalization. Decreasing q to 0.6, both the training accuracy on correctly labeled data and the test accuracy increase quickly at the early stage, but the training accuracy on wrongly labeled data also experiences a substantial rise immediately, leading to a dramatic drop in the test accuracy.

Considering both the above views, it is clear that there exists a mismatch between the statics of robust loss functions and the dynamics of DNNs learning with label noise, leading to an inferior performance. Specifically, with a static trade-off between fitting ability and robustness, the classification accuracy fails to both rise quickly at the early stage and grow steadily afterwards. As shown in Figure 1, although GCE with $q = 0.7$ achieves the highest test accuracy finally, the figure still experiences a remarkable drop at the later stage of the training process. If we slightly improve robustness by increasing q to 0.8, although the test accuracy grows steadily throughout the training process, it always remains at a lower level. By contrast, if we slightly improve fitting ability by decreasing q to 0.6, although the test accuracy reaches a higher level quickly at the early stage, it drops more dramatically afterwards.

Fortunately, a loss function with a dynamic trade-off between fitting ability and robustness can solve the above problem. Specifically, according to DNNs' learning status at different time steps, we make fitting ability stronger initially such that the classification accuracy rises quickly at the early stage, then we gradually increase the weight of robustness to make sure a steady performance growth afterwards. Moreover, to further reduce the negative impact of label noise, we let DNNs put more emphasis on easy examples than hard ones at the later stage because the easy examples are more likely to be those with correct labels (Han et al., 2018; Yu et al., 2019). We exhibit the performance of GCE with a dynamic q (DGCE) in Figure 1. At the early stage, the test accuracy of DGCE is comparable to that of GCE with $q = 0.6$. At the later stage, the figure grows steadily as GCE with $q = 0.8$. Therefore, it outperforms GCE by a remarkable margin finally.

In summary, to solve the mismatch between the static robust loss functions and the dynamic learning status of DNNs, in this paper, we propose a dynamic loss function which provides strong fitting ability initially and gradually improves robustness afterwards. The rest of the paper is organized as follows. In Section 2, we give a brief review of related work on learning with label noise. In Section 4, we introduce our proposed DGCE in detail and analyze why it achieves better performance. In Section 5, we provide extensive experimental results on various benchmark datasets. Finally, we conclude the paper in Section 6.

2 Related Work

In this section, we briefly review related work on learning with label noise.

Noise transition matrix estimation In theory, the clean class posterior can be inferred by combining the noisy class posterior and the noise transition matrix that reflects the label flipping process. With an accurately estimated noise transition matrix, one can build statistically consistent classifiers, which converge to the optimal classifiers defined by using clean data. Considering that a large estima-

tion error of the noise transition matrix would degenerate the classification accuracy significantly, numerous studies (Menon et al., 2015; Patrini et al., 2017; Yao et al., 2020; Yang et al., 2022; Zhu et al., 2022) focused on reduce this error.

Loss adjustment These methods adjust the loss of each training example before back-propagation, which could be further divided into loss reweighting (Liu & Tao, 2015; Jiang et al., 2018; Shu et al., 2020; Zhang et al., 2021b) and label correction (Tanaka et al., 2018; Yi & Wu, 2019; Song et al., 2019; Huang et al., 2020; Wang et al., 2021). The former assigns smaller weights to the potentially incorrect labels, which is usually realized by meta-learning that trains a meta DNN on a clean dataset to assign weights to each sample. The latter uses model predictions to correct the provided labels.

Sample selection These approaches attempted to select correctly labeled examples from a noisy training dataset. While small-loss trick is widely used for selecting clean labels, some recent studies (Song et al., 2021; Xia et al., 2022; Wang et al., 2022) proposed more advanced approaches. After selecting correct labels, some approaches (Han et al., 2018; Yu et al., 2019; Wei et al., 2020) directly remove wrong labeled examples and train DNNs on the remained data, while others (Nguyen et al., 2020; Li et al., 2020; Zhou et al., 2021a) only discard wrong labels but preserve the corresponding instances, then they leverage semi-supervised learning to train DNNs. To reduce the accumulated error caused by incorrect selection, this type of approaches usually maintains multiple DNNs or refines the selected set iteratively.

Regularization Many regularization methods introduce regularizers into loss functions. To name a few, Liu & Guo (2020) randomly pair an instance and another label to construct a peer sample, then uses a peer regularizer to punish DNNs from overly agreeing with the peer sample. Generalized Jenson-Shannon Divergence (GJS) (Englesson & Azizpour, 2021) introduces a consistency regularizer forcing DNNs to make consistent predictions given two augmented versions of a single input. Early learning regularization (ELR) (Liu et al., 2020) encourages the predictions of DNNs to agree with the exponential moving average of the past outputs. There are also other types of regularizations such as data augmentation (Zhang et al., 2018), gradient clipping (Menon et al., 2020), model pruning (Xia et al., 2021), over-parameterization (Liu et al., 2022), and so on.

Robust loss function While the commonly used CE easily overfits label noise due to its strong fitting ability, MAE is theoretically noise-tolerant (Ghosh et al., 2017) but suffers from underfitting due to its poor fitting ability. Subsequently, a large amount of work improves generalization performance by reconciling fitting ability with robustness in different ways. While GCE (Zhang & Sabuncu, 2018) is an interpolation between CE and MAE, symmetric cross entropy (SCE) (Wang et al., 2019b) equals a convex combination of CE and MAE, and active passive loss (Ma et al., 2020) just replaces CE in SCE with normalized CE. Taylor cross entropy (Feng et al., 2021) realizes an interpolation between CE and MAE through Taylor Series, while the work of (Englesson & Azizpour, 2021) scales the Jensen-Shannon Divergence to construct the interpolation. Robust loss functions typically cause no changes to the training process, require no extra information such as the noise rate or a clean validation set, and incur no additional memory burden or computational cost.

3 PRELIMINARIES

Risk minimization Denote the feature space by $\mathcal{X} \subset \mathbb{R}^d$ and the class space by $\mathcal{Y} = [k] = \{1, ...k\}$ where $k \geq 2$. In a typical classfication problem without label noise, the training data $\{\mathbf{x}_i, y_i\}_{i=1}^N$ is drawn i.i.d. from an unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$. The classifier $\arg\max_i f(\cdot)_i$ is a function that maps feature space to class space, where $f : \mathcal{X} \to \mathcal{C}, \mathcal{C} \subseteq [0, 1]^k, \forall \mathbf{c} \in \mathcal{C}, \mathbf{1}^T\mathbf{c} = 1$. Generally, f is a DNN with a softmax output layer. For brevity, we call f as the classifier in the following. Given a loss function $L : \mathcal{C} \times \mathcal{Y} \to \mathbb{R}_+$ and a classifier f, the L-risk of f is defined as

$$R_L(f) = \mathbb{E}_{\mathcal{D}}[L(f(\mathbf{x}), y)] = \mathbb{E}_{\mathbf{x},y}[L(f(\mathbf{x}), y)], \tag{1}$$

where $\mathbb{E}$ represents expectation. Under the risk minimization framework, our objective is to learn $f^* = \arg\min_{f \in \mathcal{H}} R_L(f)$ where $\mathcal{H}$ refers to the hypothesis space.

In the presence of label noise, we can only access the noisy training data $\{\mathbf{x}_i, \tilde{y}_i\}_{i=1}^N$ drawn i.i.d. from a noisy distribution $\tilde{\mathcal{D}}$. In this case, the L-risk of f is defined as

$$\tilde{R}_L(f) = \mathbb{E}_{\tilde{\mathcal{D}}}[L(f(\mathbf{x}), \tilde{y})] = \mathbb{E}_{\mathbf{x},\tilde{y}}[L(f(\mathbf{x}), \tilde{y})]. \tag{2}$$

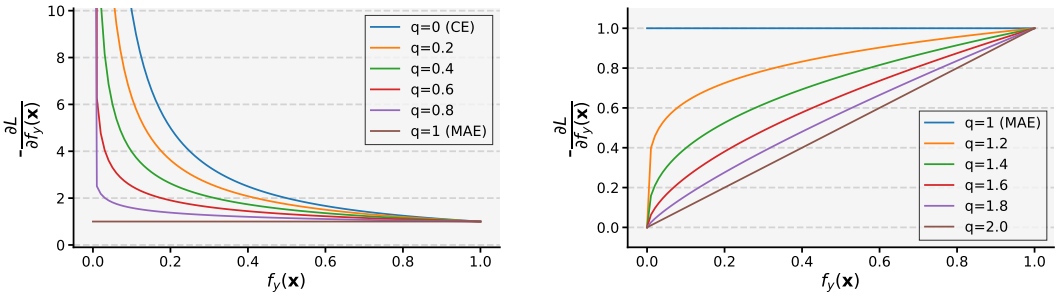

Figure 2: Gradient of GCE w.r.t the posterior probability. The left subfigure shows that when $q = 0$, GCE is equivalent to CE putting more weights on hard examples. When $q = 1$, GCE becomes MAE treating all examples equally. When $0 < q < 1$, GCE is an interpolation between CE and MAE, which puts less emphasis on hard examples compared to CE, yet still pays more attention to them compared to MAE. The right subfigure exhibits that when $q > 1$, GCE assigns more weights to easy examples instead.

Similarly, we denote the global minimizer of $\tilde{R}_L(f)$ by $\tilde{f}^*$.

Label noise model The label noise model is formulated as

$$\tilde{y}_i = \begin{cases} y_i & \text{with probability } (1 - \eta_{\mathbf{x}_i}) \\ j, j \in [k], j \neq y_i & \text{with probability } \eta_{\mathbf{x}_i,j} \end{cases}, \tag{3}$$

where $\eta_{\mathbf{x}_i} = \sum_{j \neq y_i} \eta_{\mathbf{x}_i,j}$ is called noise ratio of $\mathbf{x}_i$. This formulation corresponds to the most generic label noise termed instance-dependent noise, where the noise ratio depends on both features and labels. A special case is the asymmetric noise, where the noise ratio depends only on labels. In this case, we write $\eta_{\mathbf{x}_i} = \eta_{y_i}, \eta_{\mathbf{x}_i,j} = \eta_{y_i,j}$. In addition, the most ideal label noise is called symmetric noise, where each true label is flipped into other labels with equal probability. Formally, for symmetric noise we have $\eta_{\mathbf{x}_i} = \eta$ where η is a constant and $\eta_{\mathbf{x}_i,j} = \frac{\eta}{k-1}, \forall j \neq y_i$.

Robust loss functions The commonly used CE and MAE can be represented as

$$L_{\text{CE}}(f(\mathbf{x}), y) = -\log f_y(\mathbf{x}), \tag{4}$$

$$L_{\text{MAE}}(f(\mathbf{x}), y) = 1 - f_y(\mathbf{x}). \tag{5}$$

Ghosh et al. (2017) has proved that under symmetric label noise with $1 - \eta > \frac{1}{k}$, or asymmetric label noise with $1 - \eta_y > \eta_{y,j}, \forall j \neq y$ and $R_L(f^*) = 0$, MAE is noise-tolerant, i.e., $R_L(f^*) = R_L(\tilde{f}^*)$. However, MAE suffers from servere underfitting on complicated datasets in practice due to its poor fitting ability. Generalized cross entropy (GCE) is the most representative method among robust loss functions, which is defined as

$$L_{\text{GCE}}(f(\mathbf{x}), y) = \frac{1 - f_y^q(\mathbf{x})}{q}, \tag{6}$$

where $q \in (0, 1)$. GCE is an interpolation between CE and MAE, it becomes MAE when $q = 1$ while it is equivalent to CE when $q \to 0$ based on box-cox transformation (Atkinson et al., 2021).

4 METHOD

Gradient analysis Following Zhang & Sabuncu (2018); Feng et al. (2021), we first derive the gradients of CE, MAE, and GCE w.r.t the model parameters to demonstrate how their learning processes differ from each other. Their gradients can be represented as

$$\frac{\partial L_{\text{CE}}(f(\mathbf{x}), y)}{\partial \boldsymbol{\theta}} = -\frac{1}{f_y(\mathbf{x})} \nabla_{\boldsymbol{\theta}} f_y(\mathbf{x}), \tag{7}$$

$$\frac{\partial L_{\text{MAE}}(f(\mathbf{x}), y)}{\partial \boldsymbol{\theta}} = -\nabla_{\boldsymbol{\theta}} f_y(\mathbf{x}), \tag{8}$$

$$\frac{\partial L_{\text{GCE}}(f(\mathbf{x}), y)}{\partial \boldsymbol{\theta}} = -f_y^{q-1}(\mathbf{x}) \nabla_{\boldsymbol{\theta}} f_y(\mathbf{x}), \tag{9}$$

where $\boldsymbol{\theta}$ is the set of parameters of f. A smaller $f_y(\mathbf{x})$ means that the model prediction on $\mathbf{x}$ is less consistent with the given labels, so the loss value of $(\mathbf{x}, y)$ is larger, i.e., $(\mathbf{x}, y)$ is a harder example. As shown in the left subfigure of Figure 2, CE puts more weights on hard examples, which leads to serious overfitting in the presence of label noise, since hard examples may exactly be those with wrong labels. By contrast, MAE treats all examples equally, which avoids overfitting effectively but suffers from severe underfitting. GCE is equivalent to CE if $q = 0$ and becomes MAE if $q = 1$. When $q \in (0, 1)$, compared with CE, GCE puts less emphasis on hard examples. Relative to MAE, GCE still pays more attention to hard examples.

Dynamic GCE Although the recently proposed robust loss functions vary from each other in formulation, the common belief behind them is to reconcile fitting ability with robustness. With a static trade-off between the two, their classification accuracy fails to both rise quickly at the early stage and grow steadily at the later stage, which hinders them from achieving better generalization. Arpit et al. (2017) empirically proved that DNNs first memorize correct labels and subsequently memorize wrong labels. Therefore, it may solve the dilemma to provide a stronger fitting ability at the early stage, and then increase the weight of robustness gradually. Following this principle, we dynamically adjust q according to DNNs' learning status at different time steps. Formally, inspired by the commonly used cosine annealing learning rate schedule, we define $q(t)$ as

$$q(t) = q_s + \frac{1}{2}(q_e - q_s)(1 - \cos(\frac{\pi t}{T})), \tag{10}$$

where t and T respectively denote the current and total epoch, q_s and q_e respectively represent the start and end value of $q(t)$. Intuitively, Equation (10) represents a cosinusoidal increase from q_s to q_e. We use cosine annealing since it makes q stay relatively stable at the outset and the end of the training process.

Range of q Based on the gradient analysis, smaller q provides stronger fitting ability. Consequently, we set q_s to a small value for a quick performance rise at the early stage. Moreover, we think the range of q should not be limited within (0,1) and let $q_e > 1$. As shown in the right subfigure of Figure 2, when $q > 1$ GCE puts more emphasis on easy examples. According to the widely-used small-loss trick, the labels of easy examples are more likely to be correct. Therefore, DGCE with $q_e > 1$ can further reduce the negative impact of label noise and guarantee a steady performance growth at the later stage. Inspired by Wang et al. (2019a) which reveal the connection between loss functions and example weighting, we find that GCE with $q > 1$ plays a similar role to some reweighting methods (Majidi et al., 2021; Kumar & Amid, 2021). The main difference is that they explicitly assign more weights to potentially correctly labeled examples while DGCE implicitly realizes reweighting based on the agreement between model predictions and given labels.

Theoretical Analysis In the following, we theoretically prove that under risk minimization framework, GCE with $q > 1$ can obtain noise tolerance even under instance-dependent noise.

Definition. *The noisy loss $\tilde{L}(f(\mathbf{x}), y)$ is defined as*

$$\tilde{L}(f(\mathbf{x}), y) = (1 - \eta_{\mathbf{x}})L(f(\mathbf{x}), y) + \sum_{j \neq y} \eta_{\mathbf{x},j} L(f(\mathbf{x}), j), \tag{11}$$

such that the L-risk of f with label noise $\tilde{R}_L(f)$ can be formulated as

$$\tilde{R}_L(f) = \mathbb{E}_{\mathbf{x},\tilde{y}}[L(f(\mathbf{x}), \tilde{y})] = \mathbb{E}_{\mathbf{x},y}\mathbb{E}_{\tilde{y}|\mathbf{x},y}[L(f(\mathbf{x}), \tilde{y})] = \mathbb{E}_{\mathbf{x},y}[\tilde{L}(f(\mathbf{x}), y)]. \tag{12}$$

Lemma. *$\forall\, q > 1$ and $(\mathbf{x}, y)$, under instance-dependent noise with $1 - \eta_{\mathbf{x}} > \eta_{\mathbf{x},j}, \forall j \neq y$,*

$$\arg\min_{f(\mathbf{x}) \in \mathcal{C}} \tilde{L}(f(\mathbf{x}), y) = \arg\min_{f(\mathbf{x}) \in \mathcal{C}} L(f(\mathbf{x}), y) = \mathbf{e}_y, \tag{13}$$

where $\mathbf{e}_y$ denotes a one-hot vector with $\mathbf{e}_{yj} = 1$ if $j = y$.

Proof. Since $q > 1$ and $1 - \eta_{\mathbf{x}} > \eta_{\mathbf{x},j}$, we have

$$(1 - \eta_{\mathbf{x}}) \geq (1 - \eta_{\mathbf{x}}) \sum_j f_j^q(\mathbf{x}) \geq (1 - \eta_{\mathbf{x}}) f_y^q(\mathbf{x}) + \sum_{j \neq y} \eta_{\mathbf{x},j} f_j^q(\mathbf{x}) = 1 - q\tilde{L}(f(\mathbf{x}), y), \tag{14}$$

with equality holds iff $f(\mathbf{x}) = \mathbf{e}_y$.

Table 1: Test accuracies (%) on CIFAR-10 datasets with different levels of label noise. The best results are boldfaced while the second best results are underlined.

Method	Publication	# Hyper-parameters	Symmetric				Asymmetric		Instance	
			20%	40%	60%	80%	20%	40%	20%	40%
CE		0	83.30±0.19	67.85±0.53	47.79±0.42	25.80±0.20	86.00±0.15	74.98±0.09	80.86±0.11	61.40±0.28
GCE	NeurIPS 2018	1	90.68±0.08	87.33±0.15	81.29±0.28	61.93±0.24	88.96±0.15	58.79±0.14	89.40±0.17	78.60±3.67
SCE	ICCV 2019	2	89.24±0.43	85.37±0.36	79.20±0.26	55.65±1.36	88.05±0.07	77.38±0.15	87.91±0.19	78.62±0.64
NLNL	ICCV 2019	1	82.25±0.16	72.66±0.08	59.97±1.33	44.03±0.51	84.94±0.17	**82.13±0.03**	81.58±0.41	69.80±0.73
BTL	NeurIPS 2019	2	89.93±0.30	78.22±0.24	58.00±0.20	29.01±0.50	86.51±0.14	74.58±0.43	86.58±0.41	64.72±0.74
NCE+RCE	ICML 2020	2	90.37±0.14	87.13±0.28	80.77±0.41	64.84±0.66	88.64±0.29	76.53±0.22	89.11±0.05	77.39±0.04
TCE	IJCAI 2020	1	90.50±0.11	86.30±0.24	77.42±0.13	46.91±0.08	87.87±0.21	58.07±0.28	88.88±0.18	73.15±0.26
NCE+AGCE	ICML 2021	4	90.49±0.09	87.24±0.08	80.98±0.20	66.47±1.12	89.05±0.32	76.67±0.24	89.23±0.37	78.96±0.29
CE+SR	ICCV 2021	5	**91.40±0.13**	**88.03±0.19**	82.82±0.21	**71.34±0.45**	88.19±0.36	75.33±0.23	87.96±0.09	68.36±0.32
JS	NeurIPS 2021	1	90.62±0.06	86.28±0.28	77.04±0.54	43.04±0.34	88.30±0.16	68.73±0.50	88.27±0.56	73.92±0.33
Poly-1	ICLR 2022	1	84.89±0.08	68.25±0.67	47.87±0.56	25.14±0.32	86.19±0.20	75.52±0.42	81.37±0.23	61.50±0.60
DGCE		2	90.80±0.18	88.02±0.07	**83.11±0.27**	66.04±0.39	**89.74±0.14**	72.14±0.30	**90.02±0.15**	**83.68±0.17**

Table 2: Test accuracies (%) on CIFAR-100 datasets with different levels of label noise. The best results are boldfaced while the second best results are underlined.

Method	Publication	# Hyper-parameters	Symmetric				Asymmetric		Instance	
			20%	40%	60%	80%	20%	40%	20%	40%
CE		0	60.41±0.19	43.78±0.44	25.03±0.13	8.45±0.32	60.89±0.25	43.74±0.51	60.73±0.36	45.31±0.36
GCE	NeurIPS 2018	1	68.37±0.27	61.94±0.44	49.91±0.25	22.22±0.51	62.21±0.32	41.19±0.89	67.22±0.28	54.37±0.29
SCE	ICCV 2019	2	59.60±0.25	43.54±0.69	25.66±0.35	8.61±0.12	60.78±0.46	43.81±0.70	59.79±0.18	44.62±0.41
NLNL	ICCV 2019	1	58.01±0.30	43.66±0.75	26.64±0.29	11.62±0.36	57.25±0.04	39.31±0.37	57.73±0.20	42.92±1.17
BTL	NeurIPS 2019	2	61.83±0.13	47.54±0.16	30.47±0.54	13.73±0.38	58.70±0.14	42.91±0.32	59.31±0.40	44.18±0.44
NCE+RCE	ICML 2020	2	67.04±0.24	61.15±0.41	50.43±0.25	25.44±0.26	63.39±0.19	43.55±0.37	66.09±0.25	54.16±0.31
TCE	IJCAI 2020	1	63.97±0.65	57.40±0.63	41.46±0.67	15.12±0.27	54.97±0.49	39.73±0.19	59.42±0.40	36.48±1.38
NCE+AGCE	ICML 2021	4	67.06±0.22	60.93±0.17	49.09±0.35	20.10±0.61	64.87±0.29	**46.87±0.28**	66.38±0.46	56.35±0.10
CE+SR	ICCV 2021	5	**68.84±0.23**	62.03±0.39	50.28±0.25	9.82±1.39	59.16±0.62	41.80±0.23	63.19±0.06	47.45±0.51
JS	NeurIPS 2021	1	67.58±0.52	61.01±0.31	47.95±0.38	20.03±0.27	59.67±0.89	41.23±0.50	64.44±0.81	49.12±1.11
Poly-1	ICLR 2022	1	60.13±0.16	44.20±0.65	25.84±0.15	8.44±0.27	60.81±0.22	43.63±0.60	60.76±0.45	45.65±0.03
DGCE		2	68.77±0.35	**63.88±0.57**	**54.89±0.20**	**30.05±0.10**	**65.14±0.28**	43.08±0.18	**68.51±0.38**	**57.93±0.24**

Theorem. *For any $q > 1$, under instance-dependent label noise with $1 - \eta_{\mathbf{x}} > \eta_{\mathbf{x},j}, \forall j \neq y$, if $R_L(f^*) = 0$, we have $R_L(\tilde{f}^*) = 0$.*

Proof. Since we assume that $R_L(f^*) = 0$, we have $f^*(\mathbf{x}) = \mathbf{e}_y, \forall(\mathbf{x}, y)$. Based on Lemma, we also obtain $\tilde{f}^*(\mathbf{x}) = \mathbf{e}_y, \forall(\mathbf{x}, y)$. Therefore, $R_L(\tilde{f}^*) = R_L(f^*) = 0$.

5 EXPERIMENT

5.1 SETUP

Datasets We conduct a thorough empirical evaluation on CIFAR datasets with synthetic label noise and two real-world noisy datasets Clothing1M (Xiao et al., 2015) and Webvision (Li et al., 2017). For CIFAR, the synthetic label noise includes symmetric, asymmetric (Englesson & Azizpour, 2021), and instance(-dependent) (Xia et al., 2020) noise. For Clothing1M, we follow Liu et al. (2020) who sample 2000 mini-batches (with batch size 64) from the training data ensuring that the classes of the noisy labels are balanced. For Webvision, we follow the "Mini" setting in (Jiang et al., 2018; Ma et al., 2020) which takes only the first 50 classes of the Google resized images as the training dataset. Then we evaluate the classification performance on the same 50 classes of both Webvision and ILSVRC2012 validation set.

Baselines We first compare DGCE with Cross Entropy (CE) and several state-of-the-art loss functions causing no changes to the training process: Generalized Cross Entropy (GCE) (Zhang & Sabuncu, 2018), Negative Learning for Noisy Labels (NLNL) (Kim et al., 2019), Symmetric Cross Entropy (SCE) (Wang et al., 2019b), Bi-Tempered Logistic Loss (BTL) (Amid et al., 2019), Normalized Cross Entropy with Reverse Cross Entropy (NCE+RCE) (Ma et al., 2020), Taylor Cross Entropy (Feng et al., 2021), Normalized Cross Entropy with Asymmetric Generalized Cross Entropy (NCE+AGCE) (Zhou et al., 2021b), Cross Entropy with Sparse Regularization (CE+SR) (Zhou et al., 2021c), Jensen-Shannon Divergence Loss (JS) (Englesson & Azizpour, 2021), and Poly-1 (Leng et al., 2022). Following previous work (Zhou et al., 2021b;c; Englesson & Azizpour, 2021) we do not directly compare DGCE with either other types of methods or robust loss functions requiring pretraining such as DMI (Xu et al., 2019).

Table 3: Test accuracies (%) on Clothing1M.

CE	SCE	NCE+RCE	AGCE	CE+SR	JS	DGCE
70.30±0.12	71.38±0.12	71.44±0.15	71.49±0.16	71.47±0.18	71.45±0.24	**72.00±0.20**

Table 4: Top-1 validation accuracies (%) on WebVision and ILSVRC2012 validation set.

Dataset	CE	SCE	NCE+RCE	AGCE	CE+SR	JS	DGCE
Webvision	61.13±0.74	67.35±0.17	64.92±0.41	68.47±0.53	68.41±0.43	65.60±0.46	**70.09±0.33**
ILSVRC2012	56.53±0.43	62.13±1.05	61.75±0.42	64.56±0.14	64.87±0.34	62.16±0.16	**65.41±0.25**

In addition, to verify that DGCE can provide performance boosts for other types of methods, we integrate DGCE with the following methods: mixup (Zhang et al., 2018), early learning regularization (ELR) (Liu et al., 2020), generailized Jensen-Shannon Divergence (GJS) (Englesson & Azizpour, 2021), co-teaching (Han et al., 2018), JoCoR (Wei et al., 2020), f-divergence (Wei & Liu, 2021), Self-adaptive traing (SAT) (Huang et al., 2020), and DivideMix (Li et al., 2020).

Experimental details When comparing DGCE with other robust loss functions, we use a single shared learning setup for all approaches. For CIFAR dataset, we train a ResNet18 (He et al., 2016) using SGD for 150 epochs with momentum 0.9, weight decay 10^{-4}, batch size 128, initial learning rate 0.01, and cosine learning rate annealing. We also apply typical data augmentations including random crop and horizontal flip. For Clothing1M, we train a ResNet50 (He et al., 2016) pretrained on ImageNet using SGD for 30 epochs with momentum 0.9, weight decay 1×10^{-4}, batch size 64, and initial learning rate 0.01. The learning rate is divided by 10 at the 10th and 20th epoch. For Webvision, we train a ResNet50 (He et al., 2016) using SGD for 250 epochs with nesterov momentum 0.9, weight decay 3×10^{-5}, batch size 512, and initial learning rate 0.4. The learning rate is multiplied by 0.97 after each epoch. Typical data augmentations including random crop, color jittering, and horizontal flip are applied to both clothing1M and Webvision. More details about hyper-parameter settings can be found in Appendix A.1. When integrating DGCE with other methods, for each method we use the training setup reported in official codes unless otherwise specified. For more details, please refer to the Appendix A.2. Following most recent work (Ma et al., 2020; Feng et al., 2021; Zhou et al., 2021b;c), unless otherwise sepcified, we always report accuracies of the last epoch and all results (mean±std) are reported over 3 random runs.

5.2 Comparison with robust loss functions

The experimental results on CIFAR with label noise are summarized in Table 1 and Table 2. On CIFAR-10 dataset, DGCE achieves the best performance in 4 settings and the second best performance in 2 settings. Particularly, it outperforms all counterparts under the most complex instance-dependent label noise, and the performance gap between DGCE and the second best NCE+AGCE reaches about 5% when the noise rate is 40%. The most competitive counterparts are NCE+AGCE and CE+SR, the former achieves the second best performance in 3 settings while the latter achieves the best performance in 3 settings and the second best performance in 1 setting. However, they both have far more hyper-parameters. On more complex dataset CIFAR-100, DGCE exhibits more remarkable superiority, which reaches the best performance in 6 settings and the second best performance in 1 setting. When the noise rate is high, i.e., 60% and 80% symmetric noise, DGCE overtakes the second best method NCE+RCE by more than 4%. The evaluation on the real-world noisy datasets Clothing1M and Webvision is shown in Table 3 and Table 4, where DGCE also outperforms other approaches by a clear margin.

In summary, DGCE demonstrates a significant and consistent improvement under both synthetic and realistic label noise across various datasets. The remarkable performance validates the effectiveness of the dynamic trade-off between fitting ability and robustness.

5.3 Impact of Hyper-Parameters

We summarize the performance of DGCE with different hyper-parameters in Table 5. When $q_s = q_e$, DGCE degenerates to GCE, which serves as the baseline. As shown in Table 5, DGCE with $q_e = 1$ overtakes static GCE in most cases, especially on the more complex dataset CIFAR-100, which

Table 5: Performance of DGCE with different hyper-parameters.

Dataset	(q_s, q_e)	Symmetric				Asymmetric		Instance	
		20%	40%	60%	80%	20%	40%	20%	40%
CIFAR-10	(0.9,0.9)	90.68±0.08	87.33±0.15	81.29±0.28	61.93±0.24	88.96±0.15	58.79±0.14	89.40±0.17	78.60±3.67
	(0.8,1.0)	90.72±0.06	87.05±0.02	80.01±0.25	54.63±1.41	88.86±0.05	69.47±0.06	89.07±0.14	79.04±0.49
	(0.8,2.0)	**90.80±0.18**	**88.02±0.07**	**83.11±0.27**	**66.04±0.39**	**89.74±0.14**	**72.14±0.30**	**90.02±0.15**	**83.68±0.17**
CIFAR-100	(0.7,0.7)	68.37±0.27	61.94±0.44	49.91±0.25	22.22±0.51	62.21±0.32	41.19±0.89	67.22±0.28	54.37±0.29
	(0.6,1.0)	68.75±0.26	63.59±0.16	52.38±0.22	24.16±0.49	63.95±0.30	41.95±0.30	68.24±0.29	55.04±0.69
	(0.6,1.5)	**68.77±0.35**	**63.88±0.57**	**54.89±0.20**	**30.05±0.10**	**65.14±0.28**	**43.08±0.18**	**68.51±0.38**	**57.93±0.24**

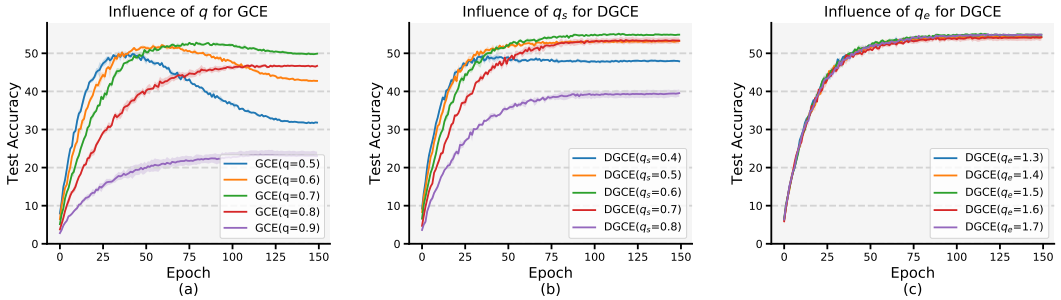

Figure 3: Hyper-parameter sensitivity of GCE and DGCE on CIFAR-100 with 60% symmetric noise.

verifies the importance of the dynamic trade-off between fitting ability and robustness. However, DGCE with $q_e = 1$ stills lags behind GCE in some settings on CIFAR-10. Since CIFAR-10 is a relatively simple dataset, $q_e = 1$ is not enough to offset the strong ability provided by a smaller q_s. By contrast, DGCE with $q_e > 1$ achieves the best performance consistently, its superiority is more remarkable with a higher noise rate.

Moreover, we investigate the hyper-parameter sensitivity of DGCE on CIFAR-100 with 60% symmetric noise. For comparison we firstly vary q for GCE from 0.5 to 0.9 (see Figure 3 (a)). Then we fix $q_e = 1.5$ and vary q_s for DGCE from 0.4 to 0.8 (see Figure 3 (b)). We finally fix $q_s = 0.6$ and vary q_e for DGCE from 1.3 to 1.7 (see Figure 3 (c)). It is clear from Figure 3 (a) that GCE is pretty sensitive to q, its test accuracy ranges from 20% to 50%. By contrast, the performance gap of DGCE with different q_s is much narrower. If we limit q_s within [0.5, 0.7], the gap is even less than 2%. Moreover, varying q_e within a reasonable range almost has no impact as shown in Figure 3 (c). Overall, DGCE is pretty robust to its hyper-parameters.

5.4 Integrating DGCE with Other Types of Algorithms

Robust loss functions have been an independent research line of handling label noise in most previous work (Ma et al., 2020; Feng et al., 2021; Zhou et al., 2021b;c). In this section, we integrate our proposed DGCE with other types of algorithms. Experimental results are summarized in Table 6. In the following, we roughly describe how we incorporate DGCE into existing frameworks. For more details please refer to Appendix A.2.

Mixup is a type of data augmentation, which can be integrated with DGCE naturally. ELR and GJS both improve robustness against label noise through regularizers. For simplicity, we directly combine DGCE with these regularizers. Co-teaching and JoCoR both select examples based on the small-loss trick during the training process, we replace CE in these frameworks with DGCE. Overall, DGCE can give large performance boosts to the above methods because its dynamic q reduces the negative impact of label noise gradually.

F-divergence is a successor to peer loss (Liu & Guo, 2020) achieving better performance. SAT uses model predictions to correct the provided labels. DivideMix is an aggregation of multiple techniques based on a sophisticated semi-supervised learning framework MixMatch (Berthelot et al., 2019). The above methods all have a warmup stage at the outset during which they train DNNs with CE without any modification. We replace CE with DGCE during the warmup stage, then for DivideMix we particularly use examples whose predictions agree with their labels to train DNNs with CE for another several epochs. Compared to CE, DGCE can provide a better initialization for the subsequent process, which can both make more reliable predictions and better discriminate correctly from wrongly labeled examples, so it also give performance gains to both of them in most cases.

Table 6: Test accuracies (%) on CIFAR-100 datasets with different levels of label noise.

Method	Symmetric				Asymmetric		Instance	
	20%	40%	60%	80%	20%	40%	20%	40%
mixup	65.90±0.32	51.21±0.61	32.71±0.14	13.82±0.07	65.79±0.67	47.31±0.27	66.18±0.14	51.02±0.40
+DGCE	**71.57±0.11**	**68.03±0.28**	**61.22±0.10**	**44.76±0.49**	**70.12±0.09**	**48.19±0.37**	**71.64±0.32**	**65.07±0.23**
ELR	72.81±0.16	69.54±0.18	62.98±0.48	29.04±0.74	**74.25±0.18**	68.41±0.11	**73.69±0.08**	72.10±0.21
+DGCE	**73.42±0.20**	**71.16±0.44**	**65.90±0.18**	**42.83±1.41**	74.14±0.15	**70.33±0.20**	73.30±0.28	**72.13±0.29**
GJS	**76.05±0.10**	72.52±0.12	63.15±0.23	**33.47±0.49**	73.98±0.32	56.83±0.37	75.46±0.23	67.53±0.35
+DGCE	75.78±0.11	**73.31±0.14**	**65.44±0.19**	32.14±0.22	**75.44±0.32**	**62.69±0.41**	**75.79±0.10**	**71.36±0.23**
Co-teaching	66.26±0.44	59.93±0.27	48.40±0.32	19.92±0.24	64.10±0.07	46.74±0.06	65.46±0.28	53.71±0.43
+DGCE	**68.00±0.30**	**62.31±0.27**	**51.78±0.27**	**27.42±0.20**	**64.45±0.10**	**48.22±0.63**	**65.97±0.09**	**54.41±0.33**
JoCoR	63.36±0.21	59.30±0.19	51.88±0.20	27.59±1.09	56.85±0.32	40.76±0.18	61.33±0.14	52.40±0.65
+DGCE	**64.83±0.23**	**60.68±0.24**	**53.59±0.44**	**32.61±0.28**	**60.20±0.40**	**41.61±0.70**	**63.09±0.49**	**52.74±0.40**
f-divergence	69.97±0.10	65.04±0.06	57.55±0.16	28.06±0.08	68.74±0.08	**54.17±0.12**	69.85±0.10	60.39±0.15
+DGCE	**71.17±0.12**	**66.74±0.06**	**59.94±0.27**	**28.72±0.08**	**70.44±0.11**	52.24±0.41	**70.93±0.07**	**61.33±0.07**
SAT	75.82±0.06	71.05±0.12	63.23±0.45	37.10±0.29	**77.57±0.07**	**70.87±0.61**	**77.19±0.40**	73.43±0.55
+DGCE	**76.32±0.12**	**73.61±0.17**	**68.37±0.68**	**41.99±0.95**	77.18±0.20	69.96±0.37	76.62±0.19	**73.89±0.13**
DivideMix	76.72±0.09	74.88±0.38	70.20±0.20	54.12±0.19	76.01±0.17	52.96±0.72	76.53±0.07	69.39±0.23
+DGCE	**77.65±0.03**	**76.14±0.10**	**71.77±0.09**	**56.67±0.14**	**76.51±0.08**	**53.30±0.51**	**77.43±0.14**	**69.87±0.56**

5.5 Robustness against Backdoor Attacks

Besides label noise, we observe that DGCE also helps to improve robustness against backdoor attacks. Typical backdoor attacks (Liu et al., 2018; Nguyen & Tran, 2021) inject triggers into a small part of training examples and change their labels into a specific target class, such that the target model performs well on benign samples whereas consistently classifies any input containing the backdoor trigger into the target class. Following Weng et al. (2020), we use two types of backdoor triggers as shown in Figure 4, which are added into 0.2% randomly selected training examples whose labels are converted into the target class. We first feed clean test samples into the target model to calculate the test accuracy, then remove test samples which are classified into the target class and add triggers into all remained test images to compute the backdoor success rate.

Figure 4: Illustration of backdoor triggers.

As shown in Table 7, test accuracies of DNNs trained with different loss functions are comparable to each other. Moreover, GCE exhibits stronger robustness against backdoor attacks than CE while DGCE can further improve backdoor robustness. Based on the dynamics of DNNs, they firstly learn patterns shared by most training examples and eventually memorize the correlation between backdoor triggers and target class because the poisoned examples only account for a small proportion (0.2%). Since DGCE reduces fitting ability gradually, it also helps to improve backdoor robustness.

Table 7: Backdoor Attack on DNNs trained with different loss functions.

Dataset	Method	Local Trigger		Global Trigger	
		Test Accuracy	Success Rate	Test Accuracy	Success Rate
CIFAR-10	CE	93.54±0.17	99.96±0.04	93.46±0.21	64.99±1.01
	GCE	92.41±0.13	0.68±0.71	92.37±0.07	38.55±2.35
	DGCE	92.28±0.20	**0.29±0.15**	92.18±0.18	**36.49±0.46**
CIFAR-100	CE	73.52±0.30	95.90±2.29	73.67±0.43	47.58±0.59
	GCE	71.32±0.20	100.00±0.00	71.81±0.17	29.45±4.35
	DGCE	72.16±0.23	**0.17±0.05**	72.01±0.24	**22.65±3.41**

6 Conclusion

In this paper, we propose a dynamic loss function DGCE to handle the mismatch between the statics of robust loss functions and dynamics of DNNs learning with label noise. At the early stage, since DNNs tend to learn generalized patterns, DGCE provides strong ability to achieve a high test accuracy quickly. Subsequently, as DNNs overfit label noise gradually, DGCE improves the weight of robustness to guarantee a steady performance growth at the later stage. Our extensive experimental results show that the simple DGCE achieves state-of-the-art performance on various benchmark datasets. Moreover, we also empirically prove that DGCE is complementary to other types of robust learning algorithms and help to improve robustness against backdoor attacks.

REFERENCES

Ehsan Amid, Manfred K Warmuth, Rohan Anil, and Tomer Koren. Robust bi-tempered logistic loss based on bregman divergences. In *Advances in Neural Information Processing Systems*, pp. 15013–15022, 2019.

Devansh Arpit, Stanislaw Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, pp. 233–242, 2017.

Anthony C Atkinson, Marco Riani, and Aldo Corbellini. The box–cox transformation: Review and extensions. *Statistical Science*, 36(2):239–255, 2021.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Avital Oliver, Nicolas Papernot, and Colin Raffel. Mixmatch: a holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 5049–5059, 2019.

Erik Englesson and Hossein Azizpour. Generalized jensen-shannon divergence loss for learning with noisy labels. In *Advances in Neural Information Processing Systems*, 2021.

Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An. Can cross entropy loss be robust to label noise? In *International Joint Conferences on Artificial Intelligence*, pp. 2206–2212, 2021.

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

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor W Tsang, and Masashi Sugiyama. Co-teaching: robust training of deep neural networks with extremely noisy labels. In *Advances in Neural Information Processing Systems*, pp. 8536–8546, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: beyond empirical risk minimization. In *Advances in Neural Information Processing Systems*, pp. 19365–19376, 2020.

Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2304–2313, 2018.

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

Abhishek Kumar and Ehsan Amid. Constrained instance and class reweighting for robust learning under label noise. *arXiv preprint arXiv:2111.05428*, 2021.

Zhaoqi Leng, Mingxing Tan, Chenxi Liu, Ekin Dogus Cubuk, Xiaojie Shi, Shuyang Cheng, and Dragomir Anguelov. Polyloss: A polynomial expansion perspective of classification loss functions. In *International Conference on Learning Representations*, 2022.

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

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

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. In *Advances in Neural Information Processing Systems*, pp. 20331–20342, 2020.

Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by over-parameterization. In *International Conference on Machine Learning*, 2022.

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

Yang Liu and Hongyi Guo. Peer loss functions: Learning from noisy labels without knowing noise rates. In *International Conference on Machine Learning*, pp. 6226–6236, 2020.

Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *ISOC Network and Distributed System Security Symposium*, 2018.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pp. 3355–3364, 2018.

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

Negin Majidi, Ehsan Amid, Hossein Talebi, and Manfred K Warmuth. Exponentiated gradient reweighting for robust training under label noise and beyond. *arXiv preprint arXiv:2104.01493*, 2021.

Aditya Menon, Brendan Van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from corrupted binary labels via class-probability estimation. In *International Conference on Machine Learning*, pp. 125–134, 2015.

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

Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021.

Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations*, 2020.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1944–1952, 2017.

Jun Shu, Qian Zhao, Keyu Chen, Zongben Xu, and Deyu Meng. Learning adaptive loss for robust learning with noisy labels. *arXiv preprint arXiv:2002.06482*, 2020.

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

Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Robust learning by self-transition for handling noisy labels. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1490–1500, 2021.

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

Xinshao Wang, Elyor Kodirov, Yang Hua, and Neil M Robertson. Derivative manipulation for general example weighting. *arXiv preprint arXiv:1905.11233*, 2019a.

Xinshao Wang, Yang Hua, Elyor Kodirov, David A Clifton, and Neil M Robertson. Proselflc: Progressive self label correction for training robust deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 752–761, 2021.

Yikai Wang, Xinwei Sun, and Yanwei Fu. Scalable penalized regression for noise detection in learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 346–355, 2022.

Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 322–330, 2019b.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13726–13735, 2020.

Jiaheng Wei and Yang Liu. When optimizing f-divergence is robust with label noise. In *International Conference on Learning Representations*, 2021.

Cheng-Hsin Weng, Yan-Ting Lee, and Shan-Hung Brandon Wu. On the trade-off between adversarial and backdoor robustness. In *Advances in Neural Information Processing Systems*, pp. 11973–11983, 2020.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. In *Advances in Neural Information Processing Systems*, pp. 7597–7610, 2020.

Xiaobo Xia, Tongliang Liu, Bo Han, Chen Gong, Nannan Wang, Zongyuan Ge, and Yi Chang. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*, 2021.

Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Jun Yu, Gang Niu, and Masashi Sugiyama. Sample selection with uncertainty of losses for learning with noisy labels. In *International Conference on Learning Representations*, 2022.

Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2691–2699, 2015.

Yilun Xu, Peng Cao, Yuqing Kong, and Yizhou Wang. Ldmi: a novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems*, pp. 6225–6236, 2019.

Shuo Yang, Erkun Yang, Bo Han, Yang Liu, Min Xu, Gang Niu, and Tongliang Liu. Estimating instance-dependent bayes-label transition matrix using a deep neural network. In *International Conference on Machine Learning*, pp. 25302–25312. PMLR, 2022.

Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. In *Advances in Neural Information Processing Systems*, pp. 7260–7271, 2020.

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

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

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021a.

HaiYang Zhang, XiMing Xing, and Liang Liu. Dualgraph: A graph-based method for reasoning about label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9654–9663, 2021b.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in Neural Information Processing Systems*, pp. 8792–8802, 2018.

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

Xiong Zhou, Xianming Liu, Junjun Jiang, Xin Gao, and Xiangyang Ji. Asymmetric loss functions for learning with noisy labels. In *International Conference on Machine Learning*, pp. 12846–12856, 2021b.

Xiong Zhou, Xianming Liu, Chenyang Wang, Deming Zhai, Junjun Jiang, and Xiangyang Ji. Learning with noisy labels via sparse regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 72–81, 2021c.

Zhaowei Zhu, Jialu Wang, and Yang Liu. Beyond images: Label noise transition matrix estimation for tasks with lower-quality features. In *International Conference on Machine Learning*, 2022.

A APPENDIX

A.1 COMPARISON WITH ROBUST LOSS FUNCTION

We randomly select 10% examples from the noisy training set as the validation set for hyper-parameter search. We slightly tune the hyper-parameters of some baseline methods for better performance under our experimental setup. For DGCE, we set q_s to $q - 0.1$ where q is the hyper-parameter of GCE or AGCE, then we tune $q_e \in \{1.2, 1.5, 1.7, 2.0\}$. The best hyper-parameters are then used to train on the full training set. The final hyper-parameters that were used to get the results are shown in Table 8 and Table 9.

Table 8: Hyper-parameters on CIFAR.

Method	Publication	Hyper-parameter	CIFAR-10	CIFAR-100
GCE	NeurIPS 2018	(q)	(0.9)	(0.7)
SCE	ICCV 2019	(α, β)	(0.1, 10.0)	(5.0, 1.0)
NLNL	ICCV 2019	(N)	(1)	(110)
BTL	NeurIPS 2019	(t_1, t_2)	(0.7, 1.5)	(0.7, 3.0)
NCE+RCE	ICML 2020	(α, β)	(1.0, 0.1)	(10.0, 0.1)
TCE	IJCAI 2020	(t)	(3)	(18)
NCE+AGCE	ICML 2021	(α, β, a, q)	(1.0, 0.4, 6, 1.5)	(10.0, 0.1, 3, 3)
CE+SR	ICCV 2021	(τ, λ_0, r, p, ρ)	(0.5, 1.5, 1, 0.1, 1.02)	(0.5, 8.0, 1, 0.01, 1.02)
JS	NeurIPS 2021	(π_1)	(0.9)	(0.5)
Poly-1	ICLR 2022	(ϵ_1)	(5)	(2)
DGCE		(q_s, q_e)	(0.8, 2,0)	(0.6, 1.5)

Table 9: Hyper-parameters on Clothing1M and Webvision.

	SCE (α, β)	NCE+RCE (α, β)	AGCE (a, q)	CE+SR (τ, λ_0, r, p, ρ)	JS (π_1)	DAL (q_s, q_e)
Clothing1M	(1.0, 1.0)	(10.0, 0.1)	(1e-5, 0.8)	(0.5, 5.0, 1, 0.01, 1.02)	(0.7)	(0.7, 1.7)
Webvision	(10.0, 1.0)	(50.0, 0.1)	(1e-5, 0.5)	(0.5, 2.0, 1, 0.01, 1.02)	(0.1)	(0.4, 1.5)

A.2 INTEGRATING DGCE WITH OTHER TYPES OF ALGORITHMS

Mixup (Zhang et al., 2018) is a type of data augmentation. To integrate DGCE with mixup, we replace CE with DGCE and let $\lambda \sim$ Beta(10,1). (q_s, q_e) for DGCE is set to (0.6, 1.5).

ELR (Liu et al., 2020) and GJS (Englesson & Azizpour, 2021) both introduce regularizers into loss funtions. To integrate DGCE with ELR, we combine DGCE with $(q_s, q_e) = (0.1, 0.5)$ and ELR with $(\lambda, \beta) = (2.0, 0.9)$. For GJS, we set (π_1, π_2, π_3) to $(0.3, 0.35, 0.35)$ on CIFAR-100, then we combine DGCE with JS Divergence as follows:

$$(1-\alpha)(1-(\frac{f(\mathbf{x}^{(1)})_y + f(\mathbf{x}^{(2)})_y}{2})^q(t)) + \alpha \text{JS}(f(\mathbf{x}^{(1)}) \| f(\mathbf{x}^{(2)})), \tag{15}$$

where $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are two versions generated from $\mathbf{x}$ by RandAugment. We set (q_s, q_e) to $(0.2, 1.0)$ and increases α from 0 to 0.6 also based on cosine annealing.

Co-teaching (Han et al., 2018) and JoCoR (Wei et al., 2020) both use small-loss trick to eliminate examples potentially with wrong labels. To integrate DGCE with them, we replace CE with DGCE and set T_k to 50. (q_s, q_e) for DGCE is set to $(0.2, 1.0)$.

F-divergence (Wei & Liu, 2021), SAT (Huang et al., 2020), and DivideMix (Li et al., 2020) all have a warmup stage at the outset during which they train DNNs with CE without any modification. To incorporate DGCE into them, we replace CE with DGCE during the warmup stage. We use Total Variation divergence in this paper, and report its test accuracies of the epoch where validation accuracy is maximum since it relies on a noisy validation set for model selection. For DivideMix, we first train DNNs with DGCE for 50 epochs, then use examples whose predictions agree with their labels to train DNNs with CE for another 10 epochs. (q_s, q_e) is set to $(0.1, 0.5)$ for f-divergence and SAT while the figures are $(0.2, 0.8)$ for DivideMix.