CONFIDENCE ADAPTIVE REGULARIZATION FOR DEEP LEARNING WITH NOISY LABELS

Anonymous authors
Paper under double-blind review

ABSTRACT

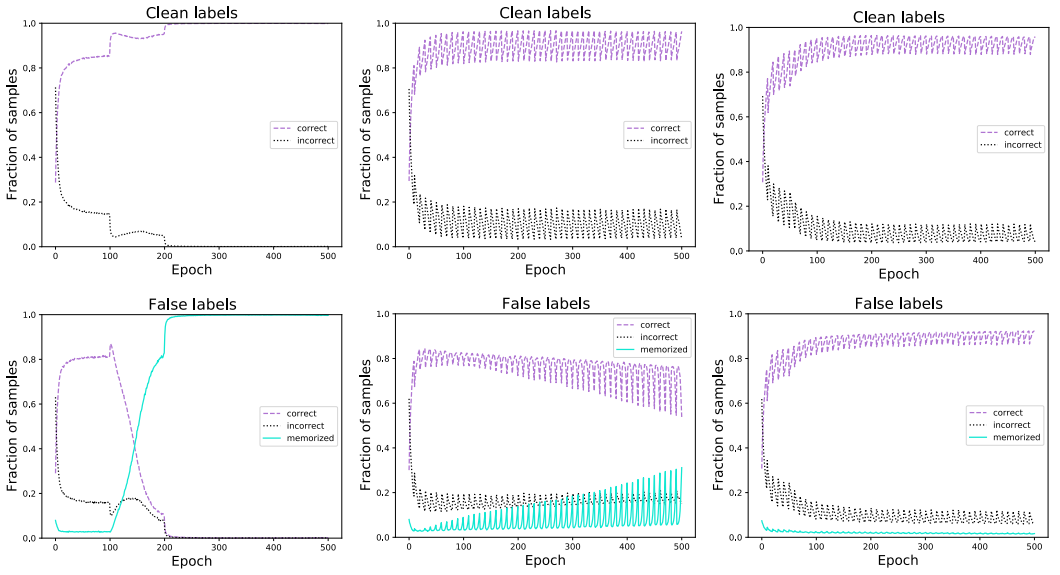
Recent studies on the memorization effects of deep neural networks on noisy labels show that the networks first fit the correctly labeled training samples before memorizing the mislabeled samples. Motivated by this *early-learning* phenomenon, we propose a novel method to prevent memorization of the mislabeled samples. Unlike the existing approaches which use confidence (captured by winning score from model prediction) to identify or ignore the mislabeled samples, we introduce an indicator branch to the original model and enable the model to produce a *new confidence* (i.e. indicates whether a sample is clean or mislabeled) for each sample. The confidence values are incorporated in the proposed loss function which is learned to assign large values to correctly-labeled samples and small values to mislabeled ones. We also discuss the limitation of our approach and propose an auxiliary regularization term to enhance the robustness of the model in challenging cases. Our empirical analysis shows that the model predicts correctly for both clean and mislabeled samples in the early learning phase. Based on the predictions in each iteration, we correct the noisy labels to steer the model towards corrected targets. Further, we provide the theoretical analysis and conduct numerous experiments on synthetic and real-world datasets, demonstrating that our approach achieves comparable and even better results to the state-of-the-art methods.

1 INTRODUCTION

With the emergence of highly-curated datasets such as ImageNet (Deng et al., 2009) and CIFAR-10 (Krizhevsky et al., 2009), deep neural networks have achieved remarkable performance on many classification tasks. However, it is extremely time-consuming and expensive to label a new large-scale dataset with high-quality annotations. Alternatively, we may obtain the dataset with lower quality annotations efficiently through online keywords queries (Li et al., 2017a) or crowdsourcing (Yu et al., 2018), but *noisy labels* are inevitably introduced consequently. Previous studies (Arpit et al., 2017; Zhang et al., 2018) demonstrate that noisy labels are problematic for overparameterized neural networks, resulting in overfitting and performance degradation. Therefore, it is essential to develop noise-robust algorithms for deep learning with noisy labels.

The authors of (Arpit et al., 2017; Li et al., 2020b; Liu et al., 2020) have observed that deep neural networks learn to correctly predict the true labels for all training samples during *early learning* stage, and begin to make incorrect predictions in *memorization* stage as it gradually memorizes the mislabeled samples (in Figure 1 (a) and (b)). In this paper, we introduce a novel regularization approach to prevent the memorization of mislabeled samples (in Figure 1 (c)). Our contributions are summarized as follows:

- We introduce an indicator branch to estimate the 'confidence' of model prediction and propose a novel loss function called confidence adaptive loss (CAL) to exploit the early-learning phase. According to the intrinsic property of early learning procedure, a large confidence value is likely to be associated with a clean sample and a small confidence value with a mislabeled one.
- We explore the limitation of CAL and propose an auxiliary regularization term forming confidence adaptive regularization (CAR) to further segregate the mislabeled samples from the clean samples in challenging cases. We develop a strategy to iteratively correct the noisy labels instead of using the noisy labels directly, allowing the model to suppress the influence of the mislabeled samples.



(a) Cross Entropy with MultiStep learning rate scheduler

(b) Cross Entropy with Cosine Annealing learning rate scheduler

(c) Confidence Adaptive Regularization with Cosine Annealing learning rate scheduler

Figure 1: We conduct the experiments on the CIFAR-10 dataset with 40% symmetric label noise using ResNet34 (He et al., 2016). The top row shows the fraction of samples with clean labels that are predicted correctly (purple) and incorrectly (black). In contrast, the bottom row shows the fraction of samples with false labels that are predicted correctly (purple), *memorized* (i.e. the prediction equals the false label, shown in blue), and incorrectly predicted as neither the true nor the labeled class (black). For samples with clean labels, all three models predict them correctly with the increasing of epochs. However, for false labels in (a), the model trained with cross-entropy loss first predicts the true labels correctly, but eventually memorizes the false labels. With the cosine annealing learning rate scheduler (Loshchilov & Hutter, 2017) in (b), the model only slows down the speed of memorizing the false labels. However, our approach shown in (c) effectively prevents memorization, allowing the model to continue learning the correctly-labeled samples to attain high accuracy on samples with both clean and false labels.

- We derive the gradients of the proposed loss functions and compare them with cross-entropy loss. Most importantly, we demonstrate that CAL has a similar effect to existing regularization approaches. It neutralizes the influence of the mislabeled samples on the gradient, and ensure the contribution from correctly labeled samples to the gradient remains dominant. We also prove the noise robustness of the auxiliary term to complete the proof for noise robustness of our approach.
- We show that the proposed approach achieves comparable and even better performance to the state-of-the-art methods on four benchmarks with different types and levels of label noise. We also perform an ablation study to evaluate the influence of different components and conduct experiments to evaluate the reliability of iterative label correction.

2 Related Work

We briefly discuss the related noise-robust methods that do not require a set of clean training data (as opposed to (Xiao et al., 2015; Vahdat, 2017; Li et al., 2017b; Hendrycks et al., 2018)) and assume the label noise is instance-independent (as opposed to (Cheng et al., 2020; Xia et al., 2020)).

Loss correction These approaches focus on correcting the loss function explicitly by estimating the noise transition matrix (Goldberger & Ben-Reuven, 2016; Patrini et al., 2017; Tanno et al., 2019).

Robust loss functions These studies develop loss functions that are robust to label noise, including $\mathcal{L}_{\text{DMI}}$ (Xu et al., 2019), MAE (Ghosh et al., 2017), GCE (Zhang & Sabuncu, 2018), SL (Wang et al., 2019) NCE (Ma et al., 2020) and TCE (Feng et al., 2020). Above two categories of methods do not utilize the early learning phenomenon.

Sample selection During the early learning stage, the samples with smaller loss values are more likely to be the correctly-labeled samples. Based on this observation, MentorNet (Jiang et al., 2018) pre-trains a mentor network for selecting small-loss samples to guide the training of the student network. Co-teaching related methods (Han et al., 2018; Yu et al., 2019; Wei et al., 2020; Lu et al., 2021) maintain two networks, and each network is trained on the small-loss samples selected by its peer network. However, their limitation is that they may eliminate numerous useful samples for robust learning. **Label correction** Tanaka et al. (2018) and Yi & Wu (2019) replace the noisy labels with soft (i.e. model probability) or hard (i.e to one-hot vector) pseudo-labels. Bootstrap (Reed et al., 2015) corrects the labels by using a convex combination of noisy labels and the model predictions. SAT (Huang et al., 2020) weigh the sample with its winning score in cross-entropy loss and updates the labels with model predictions. Arazo et al. (2019) weigh the clean and mislabeled samples by fitting a two-component Beta mixture model to loss values, and corrects the labels via convex combination as in (Reed et al., 2015). Similarly, DivideMix (Li et al., 2020a) trains two networks to separate the clean and mislabeled samples via a two-component Gaussian mixture model, and further uses MixMatch (Berthelot et al., 2019) to enhance the performance. **Regularization** Li et al. (2020b) observe that when the model parameters remain close to the initialization, gradient descent *implicitly* ignores the noisy labels. Based on this observation, they prove the gradient descent early stopping is an effective regularization to achieve robustness to label noise. Hu et al. (2019) *explicitly* add the regularizer based on neural tangent kernel (Jacot et al., 2018) to limit the distance between the model parameters to initialization. ELR (Liu et al., 2020) estimates the target probability by temporal ensembling (Laine & Aila, 2017) and adds a regularization term to cross entropy loss to avoid memorization. Other regularization techniques, such as mixup augmentation (Zhang et al., 2018b), label smoothing (Szegedy et al., 2016) and weight averaging (Tarvainen & Valpola, 2017), can enhance the performance.

Our approach is related to regularization and label correction. Compared with existing approaches (Hu et al., 2019; Liu et al., 2020), where a regularization term in loss function is necessary to resist mislabeled samples, we propose a new loss function CAL which *implicitly* boosts the gradients of correctly labeled samples and diminishes the gradients of mislabeled samples. The auxiliary regularization term in our approach is an add-on component to further enhance the robustness in challenging cases. To the best of our knowledge, our approach is the first work to obtain the confidence through an extra branch and provide the gradient analysis of it. In addition, our approach is simpler and yields comparable performance without combining other regularization techniques.

3 METHODOLOGY

This section presents a framework called confidence adaptive regularization (CAR) for robust learning from noisy labels. Our approach consists of three key elements: (1) We introduce an indicator branch to the original deep neural networks and estimate the confidence of the model predictions by exploiting the early-learning phenomenon through a confidence adaptive loss (CAL). (2) We observe the limitation of our approach and propose an auxiliary regularization term explicitly designed to further separate the confidence of clean samples and mislabeled samples in challenging cases. (3) We iteratively correct the noisy labels by incorporating the model predictions through an exponential moving average strategy.

3.1 PRELIMINARY

In this paper, we assume the label noise is instance-independent. Consider the K-class classification problem in noisy-label scenario, the ground truth label y is unavailable. We have a training set $\hat{D} = \{(\boldsymbol{x}^{[i]}, \hat{y}^{[i]})\}_{i=1}^{N}$, where $\boldsymbol{x}^{[i]}$ is an input and $\hat{y}^{[i]} \in \mathcal{Y} = \{1, \dots, K\}$ is the corresponding noisy label. We denote $\hat{\boldsymbol{y}}^{[i]} \in \{0, 1\}^K$ as one-hot vector of noisy label $\hat{y}^{[i]}$. A deep neural network model $\mathcal{N}_\theta$ (i.e. prediction branch in Figure 2 (a)) maps an input $\boldsymbol{x}^{[i]}$ to a K-dimensional logits and then feeds the logits to a softmax function $\mathcal{S}(\cdot)$ to obtain $\boldsymbol{p}^{[i]}$ of the conditional probability of each class given $\boldsymbol{x}^{[i]}$, thus $\boldsymbol{p}^{[i]} = \mathcal{S}(\boldsymbol{z}^{[i]})$, $\boldsymbol{z}^{[i]} = \mathcal{N}_\theta(\boldsymbol{x}^{[i]})$. θ denotes the parameters of the neural network and $\boldsymbol{z}^{[i]} \in \mathbb{R}^{K\times 1}$ denotes the K-dimensional logits (i.e. pre-softmax output). $\boldsymbol{z}^{[i]}$ is calculated by the fully connected layer from penultimate layer $H^{[i]} \in \mathbb{R}^{M\times 1}$. $\boldsymbol{z}^{[i]} = WH^{[i]} + \boldsymbol{b}$, where $W \in \mathbb{R}^{K\times M}$ denotes the weights and $\boldsymbol{b} \in \mathbb{R}^{K\times 1}$ denotes the bias in penultimate layer. Usually, the cross-entropy

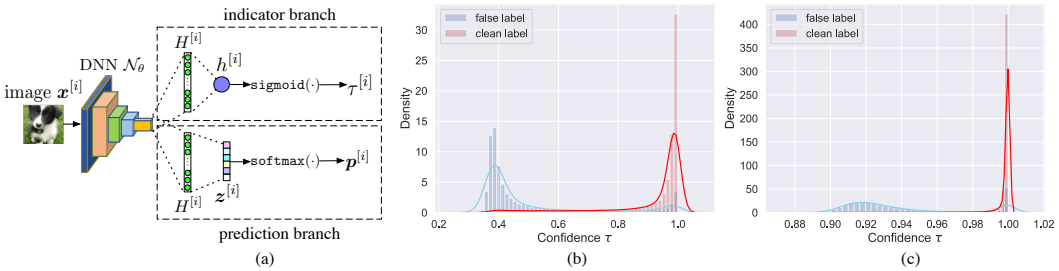


Figure 2: In (a), we introduce an indicator branch in addition to the prediction branch. Given an input image $\boldsymbol{x}^{[i]}$, the indicator branch produces a single scalar value $\tau^{[i]}$ to indicate 'confidence' and the prediction branch produces the softmax prediction probability $\boldsymbol{p}^{[i]}$. (b) and (c) show the distribution of confidence τ on the CIFAR-10 and CIFAR-100 with 40% symmetric label noise respectively.

(CE) loss reflects how well the model fits the training set $\hat{D}$:

$$\mathcal{L}_{ce} = -\frac{1}{N}\sum_{i=1}^{N}(\hat{\boldsymbol{y}}^{[i]})^T \log(\boldsymbol{p}^{[i]}). \tag{1}$$

However, as noisy label $\hat{y}^{[i]}$ is likely to be wrong, the model gradually memorizes the samples with false labels when minimizing $\mathcal{L}_{ce}$ (in Figure 1 (a) and (b)).

3.2 Confidence Adaptive Loss

In addition to the prediction branch, we introduce an indicator branch just after the penultimate layer of the original model (in Figure 2 (a)). The M-dimensional penultimate layer $H^{[i]}$ is shared in both branches. For each input $\boldsymbol{x}^{[i]}$, the prediction branch produces the softmax prediction $\boldsymbol{p}^{[i]}$ as usual. The indicator branch contains one or more fully connected layers to produce a single scalar value $h^{[i]}$, and `sigmoid` function is applied to scale it between 0 to 1. Assume we use one fully connected layer, $h^{[i]} = W'H^{[i]} + b'$, where $W' \in \mathbb{R}^{1\times M}$ denotes the weights and $b' \in \mathbb{R}$ denotes the bias in the penultimate layer of the indicator branch. Thus, we have

$$\tau^{[i]} = \texttt{sigmoid}(h^{[i]}), \quad \tau^{[i]} \in (0, 1), \tag{2}$$

where $\tau^{[i]}$ denotes the confidence value of model prediction given input $\boldsymbol{x}^{[i]}$. The early-learning phenomenon reveals that the deep neural networks memorize the correctly-labeled samples before the mislabeled samples. Thus, we assume that, *a sample with a clean label in expectation has a larger confidence value than a mislabeled sample in the early learning phase*. However, DNN model trained with CE can easily overfits to noisy labels, making the confidence (traditionally obtained by $\max_j \boldsymbol{p}_j, j \in [1, K]$) fail to capture it. To let our confidence value τ capture the above assumption, we propose the confidence adaptive cross entropy (CACE) loss

$$\mathcal{L}_{cace} = -\frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{t}^{[i]})^T \log\left(\tau^{[i]}(\boldsymbol{p}^{[i]} - \boldsymbol{t}^{[i]}) + \boldsymbol{t}^{[i]}\right), \tag{3}$$

where $\boldsymbol{t}^{[i]}$ is the one-hot vector of corrected label for each sample $\boldsymbol{x}^{[i]}$. Generally, one can directly set $\boldsymbol{t}^{[i]} = \hat{\boldsymbol{y}}^{[i]}$. However, it is less effective as $\hat{\boldsymbol{y}}^{[i]}$ can be wrong, so we propose a strategy to calculate $\boldsymbol{t}^{[i]}$ in Section 3.4. Intuitively, $\mathcal{L}_{cace}$ can be explained in two-fold: 1) In the early-learning phase, the model does not overfit the mislabeled samples. Therefore, their $\boldsymbol{p} - \boldsymbol{t}$ remain large. By minimizing $\mathcal{L}_{cace}$, it forces τ of mislabeled samples toward 0 as desired. 2) As for correctly-labeled samples, the model memorizes them first, resulting in the small $\boldsymbol{p} - \boldsymbol{t}$. Thus, it makes τ have no influence on minimizing $\mathcal{L}_{cace}$ in the case of correctly-labeled samples. As a result, by only minimizing $\mathcal{L}_{cace}$, we may obtain a trivial optimization that the model always produces $\tau \to 0$ for any inputs. To avoid this lazy learning circumstance, we introduce a penalty loss $\mathcal{L}_p$ as a cost.

$$\mathcal{L}_p = -\frac{1}{N}\sum_{i=1}^{N}\log(\tau^{[i]}), \tag{4}$$

wherein the target value of τ is always 1 for all inputs. By adding a term $\mathcal{L}_p$ to $\mathcal{L}_{cace}$, τ of correctly labeled samples are pushed to 1, and τ of mislabeled samples tend to 0 as expected. Hence, we define the confidence adaptive loss as

$$\mathcal{L}_{\text{CAL}} = \mathcal{L}_{cace} + \lambda \mathcal{L}_p, \tag{5}$$

where λ controls the strength of penalty loss. As we can see in Figure 2 (b) and (c), the confidence value τ successfully segregates the mislabeled samples from correctly-labeled samples.

3.3 AUXILIARY REGULARIZATION TERM

We observe that the early learning phenomenon is not obvious when a dataset contains too many classes (e.g. CIFAR100), i.e, the mean of τ distributions for clean samples and mislabeled samples are close to each other as shown in Figure 2 (c). Then $\mathcal{L}_{\text{CAL}}$ is likely to be reduced to $\mathcal{L}_{ce}$ (all $\tau \to 1$). To enhance the performance in this situation, we need to make τ of mislabeled samples closer to 0. Hence we propose a reverse confidence adaptive cross entropy as an auxiliary regularization term.

$$\mathcal{L}_{r\text{-}cace} = -\frac{1}{N}\sum_{i=1}^{N} \left(\tau^{[i]}(\boldsymbol{p}^{[i]} - \boldsymbol{t}^{[i]}) + \boldsymbol{t}^{[i]}\right)^T \log(\boldsymbol{t}^{[i]}). \tag{6}$$

As one-hot vector $\boldsymbol{t}^{[i]}$ is inside of the logarithm in $\mathcal{L}_{r\text{-}cace}$, this could cause computational problem when $\boldsymbol{t}^{[i]}$ contains zeros. Similar to clipping operation, we solve it by defining $\log(0) = A$ (where A is a negative constant), which will be proved important for the theoretical analysis in Section 4. Putting all together, the confidence adaptive regularization (CAR) is

$$\mathcal{L}_{\text{CAR}} = \mathcal{L}_{\text{CAL}} + \beta \mathcal{L}_{r\text{-}cace} = \mathcal{L}_{cace} + \lambda \mathcal{L}_p + \beta \mathcal{L}_{r\text{-}cace}, \tag{7}$$

where β controls the strength of regularization carried by $\mathcal{L}_{r\text{-}cace}$. In summary, $\mathcal{L}_{cace}$ is designed for learning confidence by exploiting the early-learning phenomenon. $\mathcal{L}_p$ is adopted for avoiding trivial solution. $\mathcal{L}_{r\text{-}cace}$ makes CAR robust to label noise even in challenging cases.

3.4 ITERATIVE LABEL CORRECTION

CAR requires a target probability $\boldsymbol{t}$ for each sample in the training set. Directly using the given noisy label $\hat{\boldsymbol{y}}$ as the target is less effective, since the model easily overfits to noisy labels under extreme label noise. To yield better performance, ELR (Liu et al., 2020) and SELF (Nguyen et al., 2020) use temporal ensembling (Laine & Aila, 2017) based solely on model predictions to estimate the target $\boldsymbol{t}$. However, it may lose the information of the original training set, and the predictions can be ambiguous when model overfits to noisy labels.

In this paper, we seek to iteratively correct the noisy labels for mitigating the influence of noisy labels. As shown in Figure 1, the model predicts correctly for both clean and mislabeled samples in the early learning phase. Base on this obervation, we develop a strategy to estimate the target by utilizing the noisy label $\hat{\boldsymbol{y}}$, model prediction $\boldsymbol{p}$ and confidence value τ. The target $\boldsymbol{t}^{[i]}$ of given $\boldsymbol{x}^{[i]}$ in iteration E is calculated by

$$\boldsymbol{t}^{[i]}_{[E]} = \begin{cases} \hat{\boldsymbol{y}}^{[i]} & \text{if } E < E_c \\ \alpha \boldsymbol{t}^{[i]}_{[E-1]} + (1-\alpha)\boldsymbol{p}^{[i]}_{[E]} & \text{if } E \geq E_c \text{ and } \tau^{[i]}_{[E]} \geq \delta \\ \boldsymbol{t}^{[i]}_{[E-1]} & \text{otherwise,} \end{cases} \tag{8}$$

where E_c is the iteration that starts performing label correction and $0 \leq \alpha < 1$ is the momentum. We set $E_c = 60$ by default as performance is not sensitive to the choice of E_c. Threshold δ is used to exclude ambiguous predictions with low confidence. Since we have verified that CAR only learns from the correctly labeled samples, our strategy not only enhances the stability of model predictions but also facilitates the model to learn more from clean samples. We analyze the reliability of iterative label correction and evaluate the performance with different estimation strategies in Appendix E.

4 THEORETICAL ANALYSIS

This section consists of two parts: 1) We illustrate the noise robustness of CAL by analyzing how it adjusts the gradient accordingly to achieve regularization effect. 2) We prove the robustness of the auxiliary term $\mathcal{L}_{r\text{-}cace}$ under instance-independent label noise as $\mathcal{L}_{r\text{-}cace}$ may be added to CAL.

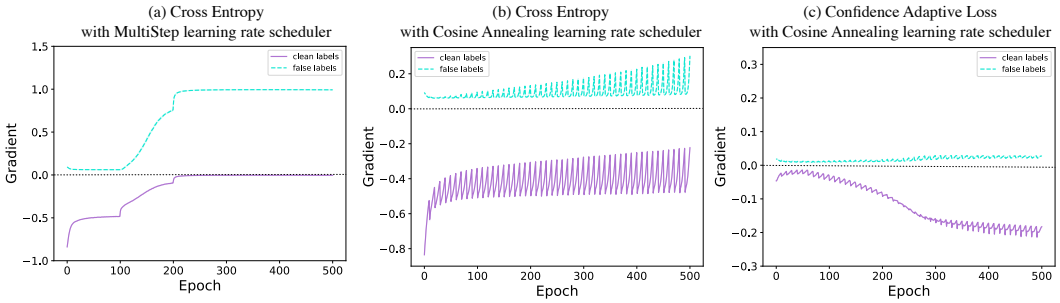


Figure 3: On CIFAR-10 with 40% symmetric label noise using ResNet34, we observe that in (a), the gradient of clean labels dominates in early learning stage, but afterwards it vanishes and the gradient of false labels dominates. In (b), it only slows down this effect with cosine annealing learning rate scheduler. In (c), CAL effectively keeps the gradient of clean labels dominant and diminishes the gradient of false labels when epoch increases, preventing memorization of mislabeled samples.

4.1 Gradient Analysis

For sample-wise analysis, we denote the true label of sample $\boldsymbol{x}$ as $y \in \{1, ..., K\}$. The ground-truth distribution over labels for sample $\boldsymbol{x}$ is $q(y|\boldsymbol{x})$, and $\sum_{k=1}^{K} q(k|\boldsymbol{x}) = 1$. Consider the case of a single ground-truth label y, then $q(y|\boldsymbol{x}) = 1$ and $q(k|\boldsymbol{x}) = 0$ for all $k \neq y$. We denote the prediction probability as $p(k|\boldsymbol{x})$ and $\sum_{k=1}^{K} p(k|\boldsymbol{x}) = 1$. For notation simplicity, we denote p_k, q_k, p_y, q_y, p_j, q_j as abbreviations for $p(k|\boldsymbol{x})$, $q(k|\boldsymbol{x})$, $p(y|\boldsymbol{x})$, $q(y|\boldsymbol{x})$, $p(j|\boldsymbol{x})$ and $q(j|\boldsymbol{x})$. Besides, we assume no label correction is performed in the following analysis.

We first explain how the cross-entropy loss $\mathcal{L}_{ce}$ (Eq. (1)) fails in noisy-label scenario. The gradient of sample-wise cross entropy loss $\mathcal{L}_{ce}$ with respect to z_j is

$$\frac{\partial \mathcal{L}_{ce}}{\partial z_j} = \begin{cases} p_j - 1 \leq 0, & q_j = q_y = 1 \\ p_j \geq 0, & q_j = 0 \end{cases} \tag{9}$$

In this case, if j is true class and equals y, but $q_j = 0$ due to the label noise, the contribution of $\boldsymbol{x}$ to the gradient is reversed. The entry corresponding to the impostor class j', is also reversed because $q_{j'} = 1$, causing the gradient of mislabeled samples dominates (in Figure 3 (a) and (b)). Thus, performing stochastic gradient descent eventually results in memorization of the mislabeled samples.

Lemma 1. *For the loss function $\mathcal{L}_{CAL}$ given in Eq. (5) and $\mathcal{L}_{CAR}$ in Eq. (7), the gradient of sample-wise $\mathcal{L}_{CAL}$ and $\mathcal{L}_{CAR}$ ($\beta = 1$) with respect to the logits z_j can be derived as*

$$\frac{\partial \mathcal{L}_{CAL}}{\partial z_j} = \begin{cases} (p_j - 1)\dfrac{p_j}{p_j - 1 + 1/\tau} \leq 0, & q_j = q_y = 1 \text{ (} j \textit{ is the true class for } \boldsymbol{x}\text{)} \quad \text{(10a)} \\ p_j \dfrac{p_y}{p_y - 1 + 1/\tau} \geq 0, & q_j = 0 \text{ (} j \textit{ is not the true class for } \boldsymbol{x}\text{)} \quad \text{(10b)} \end{cases}$$

and

$$\frac{\partial \mathcal{L}_{CAR}}{\partial z_j} = \begin{cases} (p_j - 1)\dfrac{p_j}{p_j - 1 + 1/\tau} - A\tau p_j(p_j - 1) \leq 0, & q_j = q_y = 1 \quad \text{(11a)} \\ p_j \dfrac{p_y}{p_y - 1 + 1/\tau} - A\tau p_j p_y \geq 0, & q_j = 0 \quad \text{(11b)} \end{cases}$$

respectively, where A is a negative constant defined in Section 3.3.

The proof of Lemma 1 is based on gradient derivation in two cases. We defer it in Appendix A.2.

Gradient of $\mathcal{L}_{\text{CAL}}$ in Eq. (10). Compared to the gradient of $\mathcal{L}_{ce}$ in Eq. (9), the gradient of $\mathcal{L}_{\text{CAL}}$ has an adaptive multiplier. We denote $Q = \frac{p_j}{p_j - 1 + 1/\tau}$. It is monotonically increasing on τ. We have $\lim_{\tau \to 1} Q = 1$, and $\lim_{\tau \to 0} Q = 0$. For the samples with the true class j in Eq. (10a), the cross entropy gradient term $p_j - 1$ of correctly-labeled samples tends to vanish after early learning stage

because their p_j is close to $q_j = 1$, leading mislabeled samples to dominate the gradient. However, by multiplying Q (note that $Q \to 0$ for mislabeled samples and $Q \to 1$ for correctly-labeled samples due to property of τ as we discussed in Section 3.2), it counteracts the effect of gradient dominating by mislabeled samples. For the samples that j is not the true class in Eq. (10b), the gradient term p_j is positive. Multiplying $Q < 1$ effectively dampens the magnitudes of coefficients on these mislabeled samples, thereby diminishing their effect on the gradient (in Figure 3 (c)).

Gradient of $\mathcal{L}_{\text{CAR}}$ in Eq. (11). Compared to the gradient of $\mathcal{L}_{\text{CAL}}$, an extra term derived from auxiliary regularization term $\mathcal{L}_{r\text{-}cace}$ is added. In the case of $q_j = q_y = 1$ in Eq. (11a), the extra term $-A\tau p_j(p_j - 1) < 0$ for $0 \leq p_j \leq 1$ and it is a convex quadratic function whose vertex is at $p_j = 0.5$. It means the extra term $-A\tau p_j(p_j - 1)$ provides the largest acceleration in learning around $p_j = 0.5$ where the most ambiguous scenario occurs. Intuitively, the term $-A\tau p_j(p_j - 1)$ pushes apart the peaks of confidence distribution for correctly-labeled samples and mislabeled samples. In the case of $q_j = 0$ in Eq. (11b), the extra term $-A\tau p_j p_y > 0$ is added. For correctly-labeled samples, p_y is larger, adding $-A\tau p_j p_y$ leads the residual probabilities of other unlabeled classes reduce faster. For mislabeled samples, p_y is close to 0, no acceleration needed. Overall, adding $\mathcal{L}_{r\text{-}cace}$ amplifies the effect of confidence learning in CAL, resulting in the confidence values of mislabeled samples become smaller. The empirical results of the influence of confidence distribution on CIFAR-100 with different strengths of $\mathcal{L}_{r\text{-}cace}$ are in Appendix F.

4.2 Label noise robustness

Here we prove that the $\mathcal{L}_{r\text{-}cace}$ is robust to label noise following (Ghosh et al., 2017). We assume that the noisy sample $(\boldsymbol{x}, \hat{y})$ is drawn from distribution $\mathcal{D}_\eta(\boldsymbol{x}, \hat{y})$, and the ordinary sample $(\boldsymbol{x}, y)$ is drawn from $\mathcal{D}(\boldsymbol{x}, y)$. We have $\hat{y} = i(y = i)$ with probability $\eta_{ii} = (1 - \eta)$ and $\hat{y} = j(y = i)$ with probability η_{ij} for all $j \neq i$ and $\sum_{j \neq i} \eta_{ij} = \eta$. If $\eta_{ij} = \frac{\eta}{K-1}$ for all $j \neq i$, then the noise is *uniform* or *symmetric*, otherwise, the noise is *class-conditional* or *asymmetric*. Given any classifier f and loss function $\mathcal{L}$, we define the risk of f under clean labels as $\mathcal{R}_{\mathcal{L}}(f) = \mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}[\mathcal{L}(f(\boldsymbol{x}, y))]$, and the risk under label noise rate η as $\mathcal{R}^{\eta}_{\mathcal{L}}(f) = \mathbb{E}_{\mathcal{D}(\boldsymbol{x},\hat{y})}[\mathcal{L}(f(\boldsymbol{x}, \hat{y}))]$. Let f^* and f^*_η be the global minimizers of $\mathcal{R}_{\mathcal{L}}(f)$ and $\mathcal{R}^{\eta}_{\mathcal{L}}(f)$ respectively. Then, the empirical risk minimization under loss function $\mathcal{L}$ is defined to be *noise-tolerant* if f^* is a global minimum of the noisy risk $\mathcal{R}^{\eta}_{\mathcal{L}}(f)$.

Theorem 1. *Under symmetric or uniform label noise with noise rate $\eta < \frac{K-1}{K}$, we have*

$$0 \leq \mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f^*_\eta) - \mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f^*) < \frac{-A\eta(K-1)}{K(1-\eta)-1} \tag{12}$$

and

$$A\eta < \mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*_\eta) - \mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*) \leq 0 \tag{13}$$

where f^* and f^*_η be the global minimizers of $\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f)$ and $\mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f)$ respectively.

Theorem 2. *Under class-dependent label noise with $\eta_{ij} < 1 - \eta_i, \forall j \neq i, \forall i, j \in [K]$, where $\eta_{ij} = p(\hat{y} = j | y = i), \forall j \neq i$ and $(1 - \eta_i) = p(\hat{y} = i | y = i)$, if $\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f^*) = 0$, then*

$$0 \leq \mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*) - \mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*_\eta) < G, \tag{14}$$

where $G = A(1 - K)\mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}(1 - \eta_y) > 0$, f^* and f^*_η be the global minimizers of $\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f)$ and $\mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f)$ respectively.

Due to the space constraints, we defer the proof of Theorem 1 and Theorem 2 to the Appendix A.2. Theorem 1 and Theorem 2 ensure that by minimizing $\mathcal{L}_{r\text{-}cace}$ under symmetric and asymmetric label noise, the difference of the risks caused by the derived hypotheses f^*_η and f^* are always bounded. The bounds are related to the negative constant A. Since A is the approximate of $\log(0)$ which is actually $-\infty$. A larger A (closer to 0) leads to a tighter bound but introduces a larger approximation error in implementation. A reasonable A we set is -4 in our experiments. For clarity, we also compare $\mathcal{L}_{r\text{-}cace}$ with existing noise-robust loss functions in Appendix A.3.

5 Experiments

Comparison with the state-of-the-art methods We evaluate our approach on two benchmark datasets with simulated label noise, CIFAR-10 and CIFAR-100 (Krizhevsky et al., 2009), and

Table 1: Test Accuracy (%) on CIFAR-10 and CIFAR-100 with various levels of label noise injected to the training set. We compare with previous works under the same backbone ResNet34. The results are averaged over 3 trials. Results are taken from their original papers. The best results are in **bold**. *Note that SAT, ELR and CAR use cosine annealing learning rate scheduler.*

Dataset	CIFAR-10					CIFAR-100				
Noise type	symm				asymm	symm				asymm
Method/Noise ratio	20%	40%	60%	80%	40%	20%	40%	60%	80%	40%
Cross Entropy	86.98 ± 0.12	81.88 ± 0.29	74.14 ± 0.56	53.82 ± 1.04	80.11 ± 1.44	58.72 ± 0.26	48.20 ± 0.65	37.41 ± 0.94	18.10 ± 0.82	42.74 ± 0.61
Forward $\hat{T}$ (Patrini et al., 2017)	87.99 ± 0.36	83.25 ± 0.38	74.96 ± 0.65	54.64 ± 0.44	83.55 ± 0.58	39.19 ± 2.61	31.05 ± 1.44	19.12 ± 1.95	8.99 ± 0.58	34.44 ± 1.93
Bootstrap (Reed et al., 2015)	86.23 ± 0.23	82.23 ± 0.37	75.12 ± 0.56	54.12 ± 1.32	81.21 ± 1.47	58.27 ± 0.21	47.66 ± 0.55	34.68 ± 1.10	21.64 ± 0.97	45.12 ± 0.57
GCE (Zhang & Sabuncu, 2018)	89.83 ± 0.20	87.13 ± 0.22	82.54 ± 0.23	64.07 ± 1.38	76.74 ± 0.61	66.81 ± 0.42	61.77 ± 0.24	53.16 ± 0.78	29.16 ± 0.74	47.22 ± 1.15
Joint Opt (Tanaka et al., 2018)	92.25	90.79	86.87	69.16	-	58.15	54.81	47.94	17.18	-
NLNL (Kim et al., 2019)	94.23	92.43	88.32	-	89.86	71.52	66.39	56.51	-	45.70
SL (Wang et al., 2019)	89.83 ± 0.20	87.13 ± 0.26	82.81 ± 0.61	68.12 ± 0.81	82.51 ± 0.45	70.38 ± 0.13	62.27 ± 0.22	54.82 ± 0.57	25.91 ± 0.44	69.32 ± 0.87
DAC (Thulasidasan et al., 2019)	92.91	90.71	86.30	74.84	-	73.55	66.92	57.17	32.16	-
SELF (Nguyen et al., 2020)	-	91.13	-	63.59	-	-	66.71	-	35.56	-
SAT (Huang et al., 2020)	94.14	92.64	89.23	78.58	-	75.77	71.38	62.69	**38.72**	-
ELR (Liu et al., 2020)	92.12 ± 0.35	91.43 ± 0.21	88.87 ± 0.24	**80.69 ± 0.57**	90.35 ± 0.38	74.68 ± 0.31	68.43 ± 0.42	60.05 ± 0.78	30.27 ± 0.86	73.73 ± 0.34
CAR (Ours)	**94.37 ± 0.04**	**93.49 ± 0.07**	**90.56 ± 0.07**	**80.98 ± 0.27**	**92.09 ± 0.12**	**77.90 ± 0.14**	**75.38 ± 0.08**	**69.78 ± 0.69**	**38.24 ± 0.55**	**74.89 ± 0.20**

Table 2: Comparison with state-of-the-art methods trained on Clothing1M. Results of other methods are taken from original papers. All methods use an ResNet-50 architecture pretrained on ImageNet.

CE	Forward (Patrini et al., 2017)	GCE (Zhang & Sabuncu, 2018)	SL (Wang et al., 2019)	Joint-Optim (Tanaka et al., 2018)	DMI (Xu et al., 2019)	ELR (Liu et al., 2020)	ELR+ (Liu et al., 2020)	DivideMix (Li et al., 2020a)	CAR
69.21	69.84	69.75	71.02	72.16	72.46	72.87	**74.81**	74.76	73.19

two real-world datasets, Clothing1M (Xiao et al., 2015) and WebVision (Li et al., 2017a). More information of datasets, label noise injection and training details can be found in Appendix C.

Table 1 shows the performance of CAR on CIFAR-10 and CIFAR-100 with different levels of symmetric and asymmetric label noise. All methods use the same backbone (ResNet34). We compare CAR to the state-of-the-art approaches that only modify the training loss without extra regularization techniques, such as mixup data augmentation, two networks, and weight averaging. CAR obtains the highest performance in most cases and achieves comparable results in the most challenging cases (e.g. under 80% symmetric noise). We describe the hyperparameters sensitivity of CAR in Appendix C.4. Table 2 compares CAR to state-of-the-art methods trained on the Clothing1M dataset. Note that DivideMix and ELR+ may not be completely comparable to ours as they use mixup data augmentation, two networks, and weight averaging to boost the performance, while CAR is a pure regularization method. Except for DivideMix and ELR+, CAR outperforms other methods.

Table 3: Comparison with state-of-the-art methods trained on WebVision. Results of other methods are taken from Li et al. (2020a); Liu et al. (2020). All methods use an InceptionResNetV2 architecture.

		D2L Ma et al. (2018)	MentorNet Jiang et al. (2018)	Co-teaching Han et al. (2018)	Iterative-CV Chen et al. (2019)	ELR Liu et al. (2020)	DivideMix Li et al. (2020a)	ELR+ Liu et al. (2020)	CAR
WebVision	top1	62.68	63.00	63.58	65.24	76.26	77.32	**77.78**	77.41
	top5	84.00	81.40	85.20	85.34	91.26	91.64	91.68	**92.25**
ILSVRC12	top1	57.80	57.80	61.48	61.60	68.71	**75.20**	70.29	74.09
	top5	81.36	79.92	84.70	84.98	87.84	90.84	89.76	**92.09**

Table 3 compares CAR to state-of-the-art methods trained on the mini WebVision dataset and evaluated on both WebVision and ImageNet ILSVRC12 validation sets. On WebVision, CAR outperforms others on top5 accuracy, even better than DivideMix and ELR+. On top1 accuracy, CAR is slightly superior to DivideMix and achieves comparable performance to ELR+. On ILSVRC12, DivideMix achieves superior performance in terms of top1 accuracy, while CAR achieves the best top5 accuracy even without using extra techqniues to boost the performance.

Ablation study Table 4 reports the influence of three individual components in CAR: auxiliary regularization term $\mathcal{L}_{r\text{-}cace}$, iterative label correction and indicator branch. Removing $\mathcal{L}_{r\text{-}cace}$ does not hurt the performance on CIFAR-10. However, adding the reverse term $\mathcal{L}_{r\text{-}cace}$ does improve the performance on CIFAR-100. The larger the noise is, the more improvement we obtain. Removing the

Table 4: Influence of three components in our approach. ⊖ means the model fails to converge.

Dataset	CIFAR-10			CIFAR-100		
Noise type	symm		asymm	symm		asymm
Noise ratio	40%	80%	40%	40%	80%	40%
CAR	**93.49 ± 0.07**	**80.98 ± 0.27**	**92.09 ± 0.12**	**75.38 ± 0.08**	**38.24 ± 0.55**	**74.89 ± 0.20**
– $\mathcal{L}_{r\text{-}cace}$	93.49 ± 0.07	80.98 ± 0.27	92.09 ± 0.12	74.65 ± 0.09	34.79 ± 0.71	74.73 ± 0.12
– label correction	89.47 ± 0.50	76.91 ± 0.22	88.23 ± 0.22	69.91 ± 0.21	31.33 ± 0.38	55.68 ± 0.17
– indicator branch	90.94 ± 0.28	⊖	91.55 ± 0.07	⊖	⊖	⊖

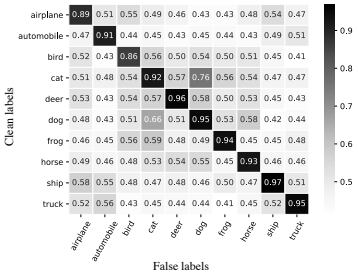


Figure 4: Average confidence values τ of false labels w.r.t clean labels on CIFAR-10 with 40% symmetric label noise.

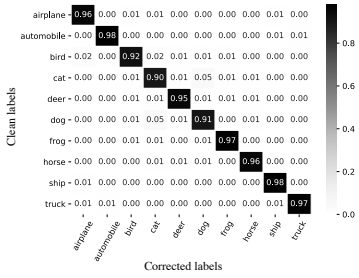


Figure 5: Confusion matrix of corrected labels w.r.t clean labels on CIFAR-10 with 40% symmetric label noise.

iterative label correction leads to a significant performance drop. This suggests that correcting the noisy labels by properly using model predictions is crucial for avoiding memorization. To validate the effect of adding the indicator branch, we conduct another way to calculate confidence value without using indicator branch: using the highest probability as the confidence value, which means $\tau^{[i]} = \max_j \boldsymbol{p}_j^{[i]}, j \in [1, K]$. Without using the indicator branch, the model only converges in two easy cases. Hence, directly calculating the confidence by model output does interfere with the original prediction branch, while adding an extra indicator branch solves this problem.

Identification of mislabeled samples When exploiting the progress of the early learning phase by CAL, we have observed that the correctly-labeled samples have larger confidence values than the mislabeled samples. We report the average confidence values of samples in Figure 4. The (i, j)-th block represents the average confidence value of samples with clean label i and false label j. We observe that the confidence values on the diagonal blocks are higher than those on non-diagonal blocks, which means that our confidence value has an effect similar to the probability of extra class in DAC (Thulasidasan et al., 2019) and AUM (Pleiss et al., 2020). The key difference is that DAC and AUM identify the mislabeled samples based on probability generated by $K+1$ class and drop the most likely mislabeled samples to perform second stage classification, while we incorporate the confidence values in loss function and implicitly achieve the regularization effect to avoid memorization of mislabeled samples. Confidence values on other levels of label noise can be found in Appendix D.

Reliability of Label correction Recall that we perform iterative label correction in Section 3.4. Since the target is calculated by a moving average between noisy labels and model predictions, our approach is able to gradually correct the false labels. The correction accuracy can be calculated by $\frac{1}{N}\sum_i^N \mathbb{1}\{\text{argmax } \boldsymbol{y}^{[i]} = \text{argmax } \boldsymbol{t}^{[i]}\}$, where $\boldsymbol{y}^{[i]}$ is the clean label of $\boldsymbol{x}^{[i]}$. We evaluate the correction accuracy on CIFAR-10 and CIFAR-100 with 40% symmetric label noise. CAR obtains correction accuracy of 95.1% and 86.4%, respectively. The confusion matrix of corrected labels w.r.t the clean labels on CIFAR-10 is shown in Figure 5. We observe that CAR corrects the false labels impressively well for all classes. Results of various levels of label noise and real-world datasets can be found in Appendix D. The evaluation for stability of iterative label correction is in Appendix G.

6 Conclusion

Based on the early learning and memorization phenomenon of deep neural networks in the presence of noisy labels, we propose an adaptive regularization method that implicitly adjusts the gradient to prevent memorization on noisy labels. Through extensive experiments across multiple datasets, our approach yields comparable and even superior results to the state-of-the-art methods.

REFERENCES

Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin Mcguinness. Unsupervised label noise modeling and loss correction. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 312–321, 2019.

Devansh Arpit, Stanisław 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 *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 233–242. JMLR. org, 2017.

David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pp. 5050–5060, 2019.

Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pp. 1062–1070, 2019.

Jiacheng Cheng, Tongliang Liu, Kotagiri Ramamohanarao, and Dacheng Tao. Learning with bounded instance and label-dependent label noise. In *International Conference on Machine Learning*, pp. 1789–1799. PMLR, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Lei Feng, Senlin Shu, Zhuoyi Lin, Fengmao Lv, Li Li, and Bo An. Can cross entropy loss be robust to label noise. In *Proceedings of the 29th International Joint Conferences on Artificial Intelligence*, pp. 2206–2212, 2020.

Aritra Ghosh, Himanshu Kumar, and PS Sastry. Robust loss functions under label noise for deep neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. 2016.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pp. 8527–8537, 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.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *NeurIPS*, 2018.

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *International Conference on Learning Representations*, 2019.

Lang Huang, Chao Zhang, and Hongyang Zhang. Self-adaptive training: beyond empirical risk minimization. *Advances in Neural Information Processing Systems*, 33, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: convergence and generalization in neural networks. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 8580–8589, 2018.

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. PMLR, 2018.

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

Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *ICLR*, 2017.

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

Mingchen Li, Mahdi Soltanolkotabi, and Samet Oymak. Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In *International Conference on Artificial Intelligence and Statistics*, pp. 4313–4324. PMLR, 2020b.

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

Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1910–1918, 2017b.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, 33, 2020.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *ICLR*, 2017.

Yangdi Lu, Yang Bo, and Wenbo He. Co-matching: Combating noisy labels by augmentation anchoring. *arXiv preprint arXiv:2103.12814*, 2021.

Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah M Erfani, Shu-Tao Xia, Sudanthi Wijewickrema, and James Bailey. Dimensionality-driven learning with noisy labels. *arXiv preprint arXiv:1806.02612*, 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. PMLR, 2020.

Tam Nguyen, C Mummadi, T Ngo, L Beggel, and Thomas Brox. Self: learning to filter noisy labels with self-ensembling. In *International Conference on Learning Representations (ICLR)*, 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.

Geoff Pleiss, Tianyi Zhang, Ethan R Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *arXiv preprint arXiv:2001.10528*, 2020.

Scott E Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. In *ICLR (Workshop)*, 2015.

Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. Prestopping: How does early stopping help generalization against label noise? 2019.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

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.

Ryutaro Tanno, Ardavan Saeedi, Swami Sankaranarayanan, Daniel C Alexander, and Nathan Silberman. Learning from noisy labels by regularized estimation of annotator confusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11244–11253, 2019.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1195–1204, 2017.

Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. Combating label noise in deep learning using abstention. In *International Conference on Machine Learning*, pp. 6234–6243. PMLR, 2019.

Arash Vahdat. Toward robustness against label noise in training deep discriminative neural networks. In *Advances in Neural Information Processing Systems*, pp. 5596–5605, 2017.

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, 2019.

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.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Parts-dependent label noise: Towards instance-dependent label noise. *arXiv preprint arXiv:2006.07836*, 2020.

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. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *NeurIPS*, pp. 6222–6233, 2019.

Kun Yi and Jianxin Wu. Probabilistic end-to-end noise correction for learning with noisy labels. In *Proceedings of the IEEE 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. PMLR, 2019.

Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. Learning with biased complementary labels. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 68–83, 2018.

C Zhang, S Bengio, M Hardt, B Recht, and O Vinyals. Understanding deep learning requires rethinking generalization, 2018.

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

Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In *Advances in neural information processing systems*, pp. 8778–8788, 2018.

A THEORETICAL ANALYSIS

A.1 GRADIENT DERIVATION OF $\mathcal{L}_{\text{CAL}}$ AND $\mathcal{L}_{\text{CAR}}$

Assume the target t equals to ground truth distribution. The sample-wise $\mathcal{L}_{\text{CAL}}$ can be rewrite as:

$$\mathcal{L}_{\text{CAL}} = \mathcal{L}_{cace} + \lambda \mathcal{L}_p = -\sum_{k=1}^{K} q_k \log(\tau(p_k - q_k) + q_k) - \lambda \log \tau. \tag{1}$$

The derivation of the $\mathcal{L}_{\text{CAL}}$ with respect to the logits is as follows:

$$\frac{\partial \mathcal{L}_{\text{CAL}}}{\partial z_j} = \frac{\partial \mathcal{L}_{cace}}{\partial z_j} = -\sum_{k=1}^{K} \frac{\tau q_k}{\tau(p_k - q_k) + q_k} \frac{\partial p_k}{\partial z_j}. \tag{2}$$

Since $p_k = \mathcal{S}(\boldsymbol{z})$=$\frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$, we have

$$\frac{\partial p_k}{\partial z_j} = \frac{\partial \left(\frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}\right)}{\partial z_j} = \frac{\frac{\partial e^{z_k}}{\partial z_j}(\sum_{j=1}^{K} e^{z_j}) - e^{z_k} \frac{\partial \left(\sum_{j=1}^{K} e^{z_j}\right)}{\partial z_j}}{(\sum_{j=1}^{K} e^{z_j})^2}. \tag{3}$$

In the case of $k = j$:

$$\begin{aligned}\frac{\partial p_k}{\partial z_j} =& \frac{\frac{\partial e^{z_k}}{\partial z_k}(\sum_{k=1}^{K} e^{z_k}) - e^{z_k} \frac{\partial \left(\sum_{k=1}^{K} e^{z_k}\right)}{\partial z_k}}{(\sum_{k=1}^{K} e^{z_k})^2} = \frac{e^{z_k}(\sum_{k=1}^{K} e^{z_k}) - e^{z_k} \cdot e^{z_k}}{(\sum_{k=1}^{K} e^{z_k})^2} \\ =& \frac{e^{z_k}}{\sum_{k=1}^{K} e^{z_k}} - \left(\frac{e^{z_k}}{\sum_{k=1}^{K} e^{z_k}}\right)^2 = p_k - p_k^2.\end{aligned} \tag{4}$$

In the case of $k \neq j$:

$$\frac{\partial p_k}{\partial z_j} = \frac{0 \cdot (\sum_{j=1}^{K} e^{z_j}) - e^{z_k} \cdot e^{z_j}}{(\sum_{j=1}^{K} e^{z_j})^2} = -\frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}} \frac{e^{z_j}}{\sum_{j=1}^{K} e^{z_j}} = -p_k p_j. \tag{5}$$

Combining Eq. (4) and (5) into Eq. (2), we obtain:

$$\begin{aligned}\frac{\partial \mathcal{L}_{\text{CAL}}}{\partial z_j} =& -\sum_{k=1}^{K} \frac{\tau q_k}{\tau(p_k - q_k) + q_k} \frac{\partial p_k}{\partial z_j} \\ =& -\frac{\tau q_j}{\tau(p_j - q_j) + q_j} \frac{\partial p_j}{\partial z_j} - \sum_{k \neq j}^{K} \frac{\tau q_k}{\tau(p_k - q_k) + q_k} \frac{\partial p_k}{\partial z_j} \\ =& -\frac{\tau q_j}{\tau(p_j - q_j) + q_j}(p_j - p_j^2) - \sum_{k \neq j}^{K} \frac{\tau q_k}{\tau(p_k - q_k) + q_k}(-p_k p_j) \\ =& -\frac{\tau q_j p_j}{\tau(p_j - q_j) + q_j} + p_j \sum_{k=1}^{K} \frac{\tau q_k p_k}{\tau(p_k - q_k) + q_k}.\end{aligned} \tag{6}$$

Therefore, if $q_j = q_y = 1$, then

$$\frac{\partial \mathcal{L}_{\text{CAL}}}{\partial z_j} = -\frac{\tau p_j}{\tau p_j - \tau + 1} + p_j \frac{\tau q_j p_j}{\tau(p_j - 1) + 1} = (p_j - 1)\frac{\tau p_j}{\tau p_j - \tau + 1} = (p_j - 1)\frac{p_j}{p_j - 1 + 1/\tau}. \tag{7}$$

If $q_j = 0$, then

$$\frac{\partial \mathcal{L}_{\text{CAL}}}{\partial z_j} = p_j \frac{\tau q_y p_y}{\tau(p_y - q_y) + q_y} = p_j \frac{p_y}{p_y - 1 + 1/\tau}. \tag{8}$$

The sample-wise $\mathcal{L}_{\text{CAR}}$ can be rewrite as (assume $\beta = 1$):

$$\mathcal{L}_{\text{CAR}} = \mathcal{L}_{\text{CAL}} + \beta\mathcal{L}_{r\text{-}cace} = \mathcal{L}_{\text{CAL}} - \sum_{k=1}^{K}(\tau(p_k - q_k) + q_k)\log q_k. \tag{9}$$

Since we have obtain the gradient of $\mathcal{L}_{\text{CAL}}$, we now only analyze the gradient of $\mathcal{L}_{r\text{-}cace}$ with respect to the logits as follows:

$$\frac{\partial\mathcal{L}_{r\text{-}cace}}{\partial z_j} = -\sum_{k=1}^{K}\frac{\tau\partial p_k}{\partial z_j}\log q_k. \tag{10}$$

Combining Eq. (4) and (5), into Eq. (10), we have

$$\begin{aligned}\frac{\partial\mathcal{L}_{r\text{-}cace}}{\partial z_j} =& -\tau(p_j - p_j^2)\log q_j - \tau\sum_{k\neq j}^{K}(-p_k p_j)\log q_k \\ =& -\tau p_j\log q_j + \tau\sum_{k=1}^{K}p_k p_j\log q_k.\end{aligned} \tag{11}$$

We denote $\log 0 = A$, thus if $q_j = q_y = 1$, then

$$\frac{\partial\mathcal{L}_{r\text{-}cace}}{\partial z_j} = -\tau p_j\log 1 + \tau p_j(p_j\log 1 + \sum_{k\neq j}^{K}p_k\log 0) = \tau p_j(1-p_j)A = -A\tau p_j(p_j - 1). \tag{12}$$

If $q_j = 0$, then

$$\frac{\partial\mathcal{L}_{r\text{-}cace}}{\partial z_j} = -\tau p_j\log 0 + \tau p_j(p_y\log 1 + (1-p_y)\log 0) = -A\tau p_j + \tau p_j(1-p_y)A = -A\tau p_j p_y. \tag{13}$$

Therefore, the gradients of $\mathcal{L}_{\text{CAR}}$ is

$$\frac{\partial\mathcal{L}_{\text{CAR}}}{\partial z_j} = \begin{cases}(p_j - 1)\frac{p_j}{p_j - 1 + 1/\tau} - A\tau p_j(p_j - 1), & q_j = q_y = 1 \\ p_j\frac{p_y}{p_y - 1 + 1/\tau} - A\tau p_j p_y, & q_j = 0\end{cases} \tag{14}$$

A.2 Formal proof for Lemma 1, Lemma2, Theorem 1 and Theorem 2

Lemma 1. *For the loss function $\mathcal{L}_{CAL}$ given in Eq. (5) and $\mathcal{L}_{CAR}$ in Eq. (7), the gradient of sample-wise $\mathcal{L}_{CAL}$ and $\mathcal{L}_{CAR}$ ($\beta = 1$) with respect to the logits z_j can be derived as*

$$\frac{\partial\mathcal{L}_{CAL}}{\partial z_j} = \begin{cases}(p_j - 1)\frac{p_j}{p_j - 1 + 1/\tau} \leq 0, & q_j = q_y = 1 \quad (j \textit{ is the true class for sample } \boldsymbol{x}) \\ p_j\frac{p_y}{p_y - 1 + 1/\tau} \geq 0, & q_j = 0 \quad (j \textit{ is not the true class for sample } \boldsymbol{x})\end{cases}$$

and

$$\frac{\partial\mathcal{L}_{CAR}}{\partial z_j} = \begin{cases}(p_j - 1)\frac{p_j}{p_j - 1 + 1/\tau} - A\tau p_j(p_j - 1) \leq 0, & q_j = q_y = 1 \\ p_j\frac{p_y}{p_y - 1 + 1/\tau} - A\tau p_j p_y \geq 0, & q_j = 0\end{cases}$$

respectively.

Proof. From the Appendix A.1, we obtain the gradient of the sample-wise $\mathcal{L}_{\text{CAL}}$ with respect to the logits z_j is

$$\frac{\partial\mathcal{L}_{\text{CAL}}}{\partial z_j} = \frac{\partial\mathcal{L}_{cace}}{\partial z_j} = -\sum_{k=1}^{K}\frac{\tau q_k}{\tau(p_k - q_k) + q_k}\frac{\partial p_k}{\partial z_j} \tag{15}$$

where $\frac{\partial p_k}{\partial z_j}$ can be further derived base on whether $k = j$ by follows:

$$\frac{\partial p_k}{\partial z_j} = \begin{cases} p_k - p_k^2 & k = j \\ -p_j p_k & k \neq j \end{cases} \tag{16}$$

According to Eq. (15) and (16), the gradient of $\mathcal{L}_{\text{CAL}}$ can be derived as:

$$\frac{\partial \mathcal{L}_{\text{CAL}}}{\partial z_j} = \begin{cases} (p_j - 1)\frac{p_j}{p_j - 1 + 1/\tau}, & q_j = q_y = 1 \\ p_j \frac{p_y}{p_y - 1 + 1/\tau}, & q_j = 0 \end{cases} \tag{17}$$

Since $p_j \leq 1$, we have $p_j - 1 \leq 0$. As $\tau < 1$, the term $\frac{p_y}{p_y - 1 + 1/\tau} > 0$, we have $(p_j - 1)\frac{p_y}{p_y - 1 + 1/\tau} \leq 0$ and $p_j \frac{p_y}{p_y - 1 + 1/\tau} \geq 0$. Similarly, the gradient of simplified $\mathcal{L}_{\text{CAR}}$ ($\beta = 1$) can be derived as:

$$\frac{\partial \mathcal{L}_{\text{CAR}}}{\partial z_j} = \frac{\partial \mathcal{L}_{\text{CAL}}}{\partial z_j} + \frac{\partial \mathcal{L}_{r\text{-}cace}}{\partial z_j} = \begin{cases} (p_j - 1)\frac{p_j}{p_j - 1 + 1/\tau} - A\tau p_j(p_j - 1), & q_j = q_y = 1 \\ p_j \frac{p_y}{p_y - 1 + 1/\tau} - A\tau p_j p_y, & q_j = 0 \end{cases} \tag{18}$$

Since A is a negative constant, we obtain $-A\tau p_j(p_j - 1) \leq 0$. Thus, in the case of $q_j = q_y = 1$, $\frac{\partial \mathcal{L}_{\text{CAR}}}{\partial z_j} \leq 0$ and in the case of $q_j = 0$, $\frac{\partial \mathcal{L}_{\text{CAR}}}{\partial z_j} \geq 0$ as claimed. Complete derivations can be found in the Appendix A.1. □

The result in Lemma 1 ensures that, during the gradient decent, learning continues on true classes when trained with $\mathcal{L}_{\text{CAL}}$ and $\mathcal{L}_{\text{CAR}}$. We then prove the noise robustness of $\mathcal{L}_{r\text{-}cace}$.

Recall that noisy label of $\boldsymbol{x}$ is $\hat{y} \in \{1, ..., K\}$ and its true label is $y \in \{1, ..., K\}$. We assume that the noisy sample $(\boldsymbol{x}, \hat{y})$ is drawn from distribution $\mathcal{D}_\eta(\boldsymbol{x}, \hat{y})$, and the ordinary sample $(\boldsymbol{x}, y)$ is drawn from $\mathcal{D}(\boldsymbol{x}, y)$. Note that this paper follows the most common setting where label noise is *instance-independent*. Then we have $\hat{y} = i(y = i)$ with probability $\eta_{ii} = (1 - \eta)$ and $\hat{y} = j(y = i)$ with probability η_{ij} for all $j \neq i$ and $\sum_{j \neq i} \eta_{ij} = \eta$. If $\eta_{ij} = \frac{\eta}{K-1}$ for all $j \neq i$, then the noise is said to be *uniform* or *symmetric*, otherwise, the noise is said to be *class-conditional* or *asymmetric*. Given any classifier f and loss function $\mathcal{L}$, we define the risk of f under clean labels as $\mathcal{R}_\mathcal{L}(f) = \mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}[\mathcal{L}(f(\boldsymbol{x}), y))]$, and the risk under label noise rate η as $\mathcal{R}^\eta_\mathcal{L}(f) = \mathbb{E}_{\mathcal{D}(\boldsymbol{x},\hat{y})}[\mathcal{L}(f(\boldsymbol{x}), \hat{y}))]$. Let f^* and f^*_η be the global minimizers of $\mathcal{R}_\mathcal{L}(f)$ and $\mathcal{R}^\eta_\mathcal{L}(f)$ respectively. Then, the empirical risk minimization under loss function $\mathcal{L}$ is defined to be *noise-tolerant* if f^* is a global minimum of the noisy risk $\mathcal{R}^\eta_\mathcal{L}(f)$.

Lemma 2. *For any $\boldsymbol{x}$, the sum of $\mathcal{L}_{r\text{-}cace}$ with respect to all the classes satisfies:*

$$0 < \sum_{j=1}^{K} \mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), j) < A(1 - K), \tag{19}$$

where $A = \log(0)$ is a negative constant that depends on the clipping operation.

Proof. By the definition of $\mathcal{L}_{r\text{-}cace}$, we can rewrite the sample-wise $\mathcal{L}_{r\text{-}cace}$ as

$$\begin{aligned} \mathcal{L}_{r\text{-}cace} &= -\sum_{k=1}^{K} \big(\tau(p(k|\boldsymbol{x}) - q(k|\boldsymbol{x})) + q(k|\boldsymbol{x})\big) \log q(k|\boldsymbol{x}) \\ &= -\big(\tau(p(y|\boldsymbol{x}) - q(y|\boldsymbol{x})) + q(y|\boldsymbol{x})\big) \log q(y|\boldsymbol{x}) - \sum\nolimits_{k \neq y} \big(\tau(p(k|\boldsymbol{x}) - q(k|\boldsymbol{x})) + q(k|\boldsymbol{x})\big) \log q(k|\boldsymbol{x}) \\ &= -\big(\tau p(y|\boldsymbol{x}) - \tau + 1\big) \log 1 - A\tau \sum\nolimits_{k \neq y} p(k|\boldsymbol{x}) \\ &= -A\tau(1 - p(y|\boldsymbol{x})). \end{aligned} \tag{20}$$

Therefore, we have

$$\sum_{j=1}^{K} \mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), j) = \sum_{j=1}^{K} -A\tau(1 - p(j|\boldsymbol{x})) = -A\tau K + A\tau \sum_{j=1}^{K} p(j|\boldsymbol{x}) = A\tau(1 - K)$$

As $\tau \in (0,1)$, A is a negative constant, K is a constant, hence

$$0 < \sum_{j=1}^{K} \mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), j) < A(1-K),$$

which concludes the proof. □

Theorem 1. *Under symmetric or uniform label noise with noise rate $\eta < \frac{K-1}{K}$, we have*

$$0 \leq \mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f_\eta^*) - \mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f^*) < \frac{-A\eta(K-1)}{K(1-\eta)-1}$$

and

$$A\eta < \mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f_\eta^*) - \mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f^*) \leq 0$$

where f^* and f_η^* be the global minimizers of $\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f)$ and $\mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f)$ respectively.

Proof. For symmetric noise, we have, for any f [1]

$$\begin{aligned}
\mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f) &= \mathbb{E}_{\mathcal{D}_\eta(\boldsymbol{x},\hat{y})}[\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}),\hat{y})] \\
&= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\mathcal{D}(y|\boldsymbol{x})}\mathbb{E}_{\mathcal{D}(\hat{y}|\boldsymbol{x},y)}[\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}),\hat{y})] \\
&= \mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}\big[(1-\eta)\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}),y) + \frac{\eta}{K-1}\sum\nolimits_{j\neq y}\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}),j)\big] \\
&= (1-\eta)\mathcal{R}_{\mathcal{L}_{r\text{-}sace}}(f) + \frac{\eta}{K-1}\Big(\sum_{j=1}^{K}\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}),j) - \mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f)\Big) \\
&= (1-\frac{\eta K}{K-1})\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f) + \frac{\eta}{K-1}\sum_{j=1}^{K}\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}),j)
\end{aligned}$$

From Lemma 2, for all f, we have:

$$\psi\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f) < \mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f) < -A\eta + \psi\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f)$$

where $\psi = (1 - \frac{\eta K}{K-1})$. Since $\eta < \frac{K-1}{K}$, we have $\psi > 0$. Thus, we can rewrite the inequality in terms of $\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f)$:

$$\frac{1}{\psi}(\mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f) + A\eta) < \mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f) < \frac{1}{\psi}\mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f)$$

Thus, for f_η^*,

$$\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f_\eta^*) - \mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f^*) < \frac{1}{\psi}(\mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f_\eta^*) - \mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f^*) - A\eta)$$

or equivalently,

$$\mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f_\eta^*) - \mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f^*) > \psi(\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f_\eta^*) - \mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f^*)) + A\eta$$

Since f^* is the global minimizer of $\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f)$ and f_η^* is the global minimizer of $\mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f)$, we have

$$0 \leq \mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f_\eta^*) - \mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f^*) < \frac{-A\eta}{\psi} = \frac{-A\eta(K-1)}{K(1-\eta)-1}$$

and

$$A\eta < \mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f_\eta^*) - \mathcal{R}^\eta_{\mathcal{L}_{r\text{-}cace}}(f^*) \leq 0$$

which concludes the proof. □

[1] In the following, note that $\mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{y|\boldsymbol{x}} = \mathbb{E}_{\boldsymbol{x},y} = \mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}$, which denote expectation with respect to the corresponding conditional distributions.

Theorem 2. *Under class-dependent label noise with $\eta_{ij} < 1 - \eta_i, \forall j \neq i, \forall i, j \in [K]$, where $\eta_{ij} = p(\hat{y} = j|y = i), \forall j \neq i$ and $(1 - \eta_i) = p(\hat{y} = i|y = i)$, if $\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f^*) = 0$, then*

$$0 \leq \mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*) - \mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*_{\eta}) < G,$$

where $G = A(1 - K)\mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}(1 - \eta_y) > 0$, f^* and f^*_{η} be the global minimizers of $\mathcal{R}_{\mathcal{L}_{r\text{-}cace}}(f)$ and $\mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f)$ respectively.

Proof. For asymmetric or class-dependent noise, we have

$$\begin{aligned}
\mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f) &= \mathbb{E}_{\mathcal{D}_{\eta}(\boldsymbol{x},\hat{y})}[\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), \hat{y})] \\
&= \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\mathcal{D}(y|\boldsymbol{x})}\Big[(1 - \eta_y)\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), y) + \sum_{j\neq y}\eta_{yj}\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), j)\Big] \\
&= \mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}\Big[(1 - \eta_y)\Big(\sum_{j=1}^{K}\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), j) - \sum_{j\neq y}\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), j)\Big)\Big] \\
&\quad + \mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}\Big[\sum_{j\neq y}\eta_{yj}\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), j)\Big] \\
&< \mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}\Big[(1 - \eta_y)\Big(A(1 - K) - \sum_{j\neq y}\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), j)\Big)\Big] \\
&\quad + \mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}\Big[\sum_{j\neq y}\eta_{yj}\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), j)\Big] \\
&= A(1 - K)\mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}(1 - \eta_y) - \mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}\Big[\sum_{j\neq y}(1 - \eta_y - \eta_{yj})\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), j)\Big].
\end{aligned}$$

On the other hand, we also have

$$\mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f) > -\mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}\Big[\sum_{j\neq y}(1 - \eta_y - \eta_{yj})\mathcal{L}_{r\text{-}cace}(f(\boldsymbol{x}), j)\Big]$$

Hence, we obtain

$$\begin{aligned}
\mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*) - \mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*_{\eta}) &< A(1 - K)\mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}(1 - \eta_y) \\
&\quad + \mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}\Big[\sum_{j\neq y}(1 - \eta_y - \eta_{yj})\Big(\mathcal{L}_{r\text{-}cace}(f^*_{\eta}(\boldsymbol{x}), j) - \mathcal{L}_{r\text{-}cace}(f^*(\boldsymbol{x}), j)\Big)\Big]
\end{aligned}$$

Next, we prove the bound. First, $(1 - \eta_y - \eta_{yj}) > 0$ as per the assumption that $\eta_{yj} < 1 - \eta_y$. Second, our assumption has $\mathcal{R}_{r\text{-}cace}(f^*) = 0$, we have $\mathcal{L}_{r\text{-}cace}(f^*(\boldsymbol{x}), y) = 0$. This is only satisfied iff $f^*_j(\boldsymbol{x}) = 1$ when $j = y$, and $f^*_j(\boldsymbol{x}) = 0$ when $j \neq y$. According to the definition of $\mathcal{L}_{r\text{-}cace}$, we have $\mathcal{L}_{r\text{-}cace}(f^*(\boldsymbol{x}), j) = -A\tau, \forall j \neq y$, and $\mathcal{L}_{r\text{-}cace}(f^*_{\eta}(\boldsymbol{x}), j) \leq -A\tau, \forall j \in [K]$. We then obtain

$$\mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}\Big[\sum_{j\neq y}(1 - \eta_y - \eta_{yj})\Big(\mathcal{L}_{r\text{-}cace}(f^*_{\eta}(\boldsymbol{x}), j) - \mathcal{L}_{r\text{-}cace}(f^*(\boldsymbol{x}), j)\Big)\Big] \leq 0$$

Therefore, we have

$$\mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*) - \mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*_{\eta}) < A(1 - K)\mathbb{E}_{\mathcal{D}(\boldsymbol{x},y)}(1 - \eta_y)$$

Since f^*_{η} is the global minimizers of $\mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f)$, we have $\mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*) - \mathcal{R}^{\eta}_{\mathcal{L}_{r\text{-}cace}}(f^*_{\eta}) \geq 0$, which concludes the proof. □

A.3 Comparison with existing noise-robust loss functions

According to the definition in Section 4, we obtain the sample-wise

$$\begin{aligned}
\mathcal{L}_{r\text{-}cace} &= -\sum_{k=1}^{K}\big(\tau(p(k|\boldsymbol{x}) - q(k|\boldsymbol{x})) + q(k|\boldsymbol{x})\big)\log q(k|\boldsymbol{x}) \\
&= -\big(\tau(p(y|\boldsymbol{x}) - q(y|\boldsymbol{x})) + q(y|\boldsymbol{x})\big)\log q(y|\boldsymbol{x}) - \sum\nolimits_{k\neq y}\big(\tau(p(k|\boldsymbol{x}) - q(k|\boldsymbol{x})) + q(k|\boldsymbol{x})\big)\log q(k|\boldsymbol{x}) \\
&= -\big(\tau p(y|\boldsymbol{x}) - \tau + 1\big)\log 1 - A\tau\sum\nolimits_{k\neq y}p(k|\boldsymbol{x}) \\
&= -A\tau(1 - p(y|\boldsymbol{x})). \text{ where } \tau \in (0, 1) \text{ and } A \text{ is a negative constant .}
\end{aligned} \tag{21}$$

Similarly, we have sample-wise $\mathcal{L}_{mae}$ Ghosh et al. (2017), $\mathcal{L}_{rce}$ Wang et al. (2019), $\mathcal{L}_{gce}$ Zhang & Sabuncu (2018) and $\mathcal{L}_{tce}$ Feng et al. (2020) as follows

$$\mathcal{L}_{mae} = \sum_{k=1}^{K} |p(k|\boldsymbol{x}) - q(k|\boldsymbol{x})| = (1 - p(y|\boldsymbol{x})) + \sum_{k \neq y} p(k|\boldsymbol{x}) = 2(1 - p(y|\boldsymbol{x}));$$

$$\mathcal{L}_{rce} = -\sum_{k=1}^{K} p(k|\boldsymbol{x}) \log q(k|\boldsymbol{x}) = -p(y|\boldsymbol{x}) \log 1 - \sum_{k \neq y} p(k|\boldsymbol{x}) \log 0 = -A(1 - p(y|\boldsymbol{x}));$$

$$\mathcal{L}_{gce} = \sum_{k=1}^{K} q(k|\boldsymbol{x}) \frac{1 - p(k|\boldsymbol{x})^{\rho}}{\rho} = q(y|\boldsymbol{x}) \frac{1 - p(y|\boldsymbol{x})^{\rho}}{\rho} = \frac{1}{\rho}(1 - p(y|\boldsymbol{x})^{\rho}), \rho \in (0, 1];$$

$$\mathcal{L}_{tce} = \sum_{i=1}^{t} \frac{(1 - p(y|\boldsymbol{x}))^{i}}{i}, t \in \mathbb{N}_{+} \text{ denotes the order of Taylor Series.}$$

We observe that when $\tau = 1$ (even though it is impossible), $\mathcal{L}_{r\text{-}cace}$ is reduced to $\mathcal{L}_{rce}$. If $A = -2$ and $\tau = 1$, $\mathcal{L}_{r\text{-}cace}$ is further reduced to $\mathcal{L}_{mae}$. Since confidence τ is various for different samples, $\mathcal{L}_{r\text{-}cace}$ is more like a dynamic version of $\mathcal{L}_{mae}$. As for $\mathcal{L}_{gce}$, $\lim_{\rho \to 0} \mathcal{L}_{gce} = \mathcal{L}_{ce}$ and $\mathcal{L}_{gce} = \frac{1}{2}\mathcal{L}_{mae}$ when $\rho = 1$. Similarly, $\lim_{t \to \infty} \mathcal{L}_{tce} = \mathcal{L}_{ce}$ and $\mathcal{L}_{tce} = \frac{1}{2}\mathcal{L}_{mae}$ when $t = 1$. Therefore, both $\mathcal{L}_{gce}$ and $\mathcal{L}_{tce}$ can be interpreted as the generalization of MAE and CE, which benefits the noise robust from MAE and training efficiency from CE. However, parameters ρ and t are fixed before training, so it is hard to tell what is the best parameter for the certain dataset. Instead, combined with $\mathcal{L}_{\text{CAL}}$, $\mathcal{L}_{r\text{-}cace}$ contains a dynamic confidence value τ for each sample that automatically learned from dataset, facilitating the learning from correctly-labeled samples.

B ALGORITHM

Algorithm 1 provides detail pseudocode for CAR. Note that for Cosine Annealing learning rate scheduler, the condition line 8 becomes $e \geq E_c \; and \; \tau^{[i]} \geq \delta \; and \; e\%E_p == 0$, where E_p is the number of epochs in each period, we fix $E_p = 10$ in all experiments.

Algorithm 1: Confidence adaptive regularization (CAR)

Input: Deep neural network $\mathcal{N}_\theta$ with trainable parameters θ; λ is the parameter for penalty term $\mathcal{L}_p$; β is the parameter for regularization term $\mathcal{L}_{r\text{-}cace}$; E_c is the epoch that starts to estimate target; α is the momentum in target estimation; training set D, batch size B, total epoch E_{max};

1 $\boldsymbol{t} = \hat{\boldsymbol{y}}$ ▷ Initialize the target by noisy labels;
2 **for** $e = 1, 2, \ldots, E_{max}$ **do**
3 **Shuffle** D into $\frac{|D|}{B}$ mini-batches ;
4 **for** $n = 1, 2, \ldots, \frac{|D|}{B}$ **do**
5 **for** i *in each mini-batch* D_n **do**
6 $\boldsymbol{p}^{[i]} = \mathcal{S}(\mathcal{N}_\theta(\boldsymbol{x}^{[i]}))$ ▷ Obtain model predictions;
7 $\tau^{[i]} = \texttt{sigmoid}(h^{[i]})$ ▷ Obtain corresponding confidence;
8 **if** $e \geq E_c$ *and* $\tau^{[i]} \geq \delta$ **then**
9 $\boldsymbol{t}^{[i]} = \alpha \boldsymbol{t}^{[i]} + (1 - \alpha)\boldsymbol{p}^{[i]}$ ▷ Iterative label correction;
10 **Calculate the loss** $\mathcal{L}_{\text{CAR}} = \mathcal{L}_{cace} + \lambda \mathcal{L}_p + \beta \mathcal{L}_{r\text{-}cace} = -\frac{1}{B}\sum_{i=1}^{B} (\boldsymbol{t}^{[i]})^T \log\left(\tau^{[i]}(\boldsymbol{p}^{[i]} - \boldsymbol{t}^{[i]}) + \boldsymbol{t}^{[i]}\right) - \frac{\lambda}{B}\sum_{i=1}^{B} \log(\tau^{[i]}) - \frac{\beta}{B}\sum_{i=1}^{B} \left(\tau^{[i]}(\boldsymbol{p}^{[i]} - \boldsymbol{t}^{[i]}) + \boldsymbol{t}^{[i]}\right)^T \log(\boldsymbol{t}^{[i]})$;
11 **Update** θ using stochatic gradient descent ;
12 **Output** θ.

Table 5: Detail information of experiment.

(a) Description of the datasets used in the experiments.

Dataset	# of train	# of val	# of test	# of classes	input size	Noise rate (%)
Datasets with clean annotation						
CIFAR-10	50K	-	10K	10	32×32	≈ 0.0
CIFAR-100	50K	-	10K	100	32×32	≈ 0.0
Datasets with real world noisy annotation						
Clothing1M	1M	14K	10K	14	224×224	≈ 38.5
Webvision 1.0	66K	-	2.5K	50	256×256	≈ 20.0

(b) Description of the hyperparameters used in our approach.

Hyperparameter	Description
λ	Control the strength of penalty loss in $\mathcal{L}_{\mathrm{CAL}}$.
β	Control the strength of regularization term $\mathcal{L}_{r\text{-}cace}$.
E_c	The epoch starts to estimate target.
α	The momentum in target estimation.
δ	The threshold of confidence in target estimation.

C Detail description of experiments

Source code for the experiments is available in the zip file. All experiments are implemented in PyTorch and run in a single Nvidia GTX 1080 GPU. For CIFAR-10 and CIFAR-100, we do not perform early stopping since we don't assume the presence of clean validation data. All test accuracy are recorded from the last epoch of training. For Clothing1M, it provides 50k, 14k, 10k refined clean data for training, validation and testing respectively. Note that we do not use the 50k clean data. We report the test accuracy when the performance on validation set is optimal. All tables of CIFAR-10/CIFAR-100 report the mean and standard deviation from 3 trails with different random seeds. As for larger datasets, we only perform a single trail.

C.1 Dataset description and preprocessing

The information of datasets are described in Table 5a. CIFAR-10 and CIFAR-100 are clean datasets, we describe the label noise injection in Appendix C.2. Clothing1M consists of 1 million training images from 14 categories collected from online shopping websites with noisy labels generated from surrounding texts. Its noise level is estimated as 38.5% (Song et al., 2019). Following (**?**Chen et al., 2019), we use the mini WebVision dataset which contains the top 50 classes from the Google image subset of WebVision, which results in approximate 66 thousand images. The noise level of WebVision is estimated at 20% (Li et al., 2017a).

As for data preprocessing, we apply normalization and regular data augmentation (i.e. random crop and horizontal flip) on the training sets of all datasets. The cropping size is consistent with existing works (Liu et al., 2020; Li et al., 2020a). Specifically, 32 for CIFAR-10 and CIFAR-100, 224×224 for Clothing 1M (after resizing to 256×256), and 227×227 for Webvision.

C.2 Simulated label noise injection

Since the CIFAR-10 and CIFAR-100 are initially clean, we follow Tanaka et al. (2018); Patrini et al. (2017) for symmetric and asymmetric label noise injection. Specifically, symmetric label noise is generated by randomly flipping a certain fraction of the labels in the training set following a uniform distribution. Asymmetric label noise is simulated by flipping their class to another certain class according to the mislabel confusions in the real world. For CIFAR-10, the asymmetric noisy labels are generated by mapping *truck* $\rightarrow$ *automobile*, *bird* $\rightarrow$ *airplane*, *deer* $\rightarrow$ *horse* and *cat* $\leftrightarrow$ *dog*. For CIFAR-100, the noise flips each class into the next, circularly within super-classes.

C.3 Training procedure

CIFAR-10/CIFAR-100: We use a ResNet-34 and train it using SGD with a momentum of 0.9, a weight decay of 0.001, and a batch size of 64. The network is trained for 500 epochs for both CIFAR-10 and CIFAR-100. We use the cosine annealing learning rate Loshchilov & Hutter (2017) where the maximum number of epoch for each period is 10, the maximum and minimum learning rate is set to 0.02 and 0.001 respectively. As for cross entropy with MultiStep learning rate scheduler in Figure 1 and Figure 3 in the paper, we set the initial learning rate as 0.02, and reduce it by a factor of 10 after 100 and 200 epochs. The reason that we train the model 500 epochs in total is to fully evaluate whether the model will overfit mislabeled samples, which avoids the interference caused by early stopping Li et al. (2020b) (i.e. the model may not start overfitting mislabeled samples when

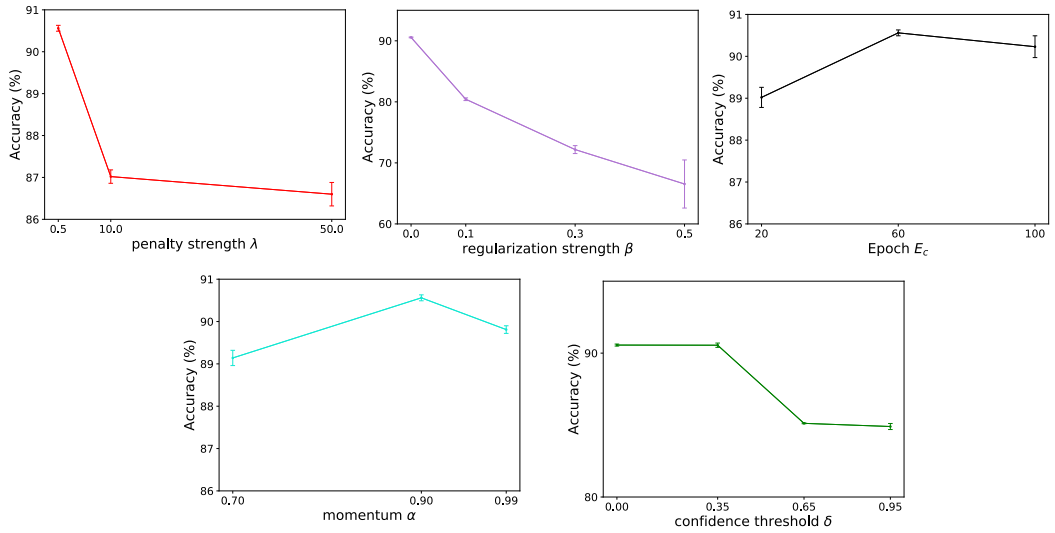


Figure 6: Test accuracy on CIFAR-10 with 60% symmetric label noise. The mean accuracy over three runs is reported, along with bars representing one standard deviation from the mean. In each experiment, the rest of hyperparameters are fixed to the values reported in Section C.4.

the number of training epochs is small, especially when learning rate scheduler is cosine annealing Loshchilov & Hutter (2017)).

Clothing1M: Following Xiao et al. (2015); Wang et al. (2019), we use a ResNet-50 pretrained on ImageNet. We train the model with batch size 64. The optimization is done using SGD with a momentum 0.9, and weight decay 0.001. We use the same cosine annealing learning rate as CIFAR-10 except the minimum learning rate is set to 0.0001 and total epoch is 400. For each epoch, we sample 2000 mini-batches from the training data ensuring that the classes of the noisy labels are balanced.

Webvision: Following Li et al. (2020a); Liu et al. (2020), we use an InceptionResNetV2 as the backbone architecture. All other optimization details are the same as for CIFAR-10, except for the weight decay (0.0005) and the batch size (32).

C.4 Hyperparameters selection and sensitivity

Table 5b provides a detailed description of hyperparameters in our approach. We perform hyperparameter tuning via grid search: $\lambda = [0.5, 10, 50]$, $\beta = [0.0, 0.1, 0.3, 0.5]$, $E_c = [20, 60, 100]$, $\alpha = [0.7, 0.9, 0.99]$ and $\delta = [0, 0, 0.35, 0.65, 0.95]$. For CIFAR-10, the selected value are $\lambda = 0.5$, $\beta = 0.0$, $E_c = 60$, $\alpha = 0.9$ and $\delta = 0.0$. For CIFAR-100 with 40% asymmetric label noise, the selected value are $\lambda = 10$, $\beta = 0.1$, $E_c = 20$, $\alpha = 0.9$, $\delta = 0.0$. For CIFAR-100 with 20%/40%/60% symmetric label noise, we set $\lambda = 10$, $\beta = 0.1$, $E_c = 60$, $\alpha = 0.9$, $\delta = 0.95$ and $\lambda = 50$, $\beta = 0.1$, $E_c = 60$, $\alpha = 0.9$, $\delta = 0.0$ for 80% symmetric label noise. For Webvision, we set $\lambda = 50$, $\beta = 0.1$, $E_c = 200$, $\alpha = 0.9$, $\delta = 0.0$. For Clothing1M, we set $\lambda = 50$, $\beta = 0.1$, $E_c = 60$, $\alpha = 0.8$, $\delta = 0.0$.

Figure 6 and Figure 7 shows the hyperparameters sensitivity of CAR on CIFAR-10 and CIFAR-100 with 60% symmetric label noise respectively. The coefficient of penalty loss λ needs to be large than 0 to avoid trivial solution but also cannot be too large for CIFAR-10, avoiding neglecting $\mathcal{L}_{cace}$ term in the loss. As the CIFAR-10 is an easy dataset, no additional regularization requires by $\mathcal{L}_{r\text{-}cace}$ term. Therefore, the regularization coefficient β should be 0 and large β may cause model to underfit. The performance is robust to E_c and α, as long as the momentum α is large enough (e.g. larger than 0.7). The choice of confidence threshold δ depends on the difficulty of dataset. A larger δ will slightly slow down the speed of label correction but helps exclude ambiguous predictions with low confidence values. Overall, the sensitivity to hyperparameters is quite mild and the performance is quite robust, unless the parameter is set to be very large or very small, resulting in neglecting $\mathcal{L}_{cace}$ term or underfitting. We can observe the similar results of CIFAR-100 in Figure 7.

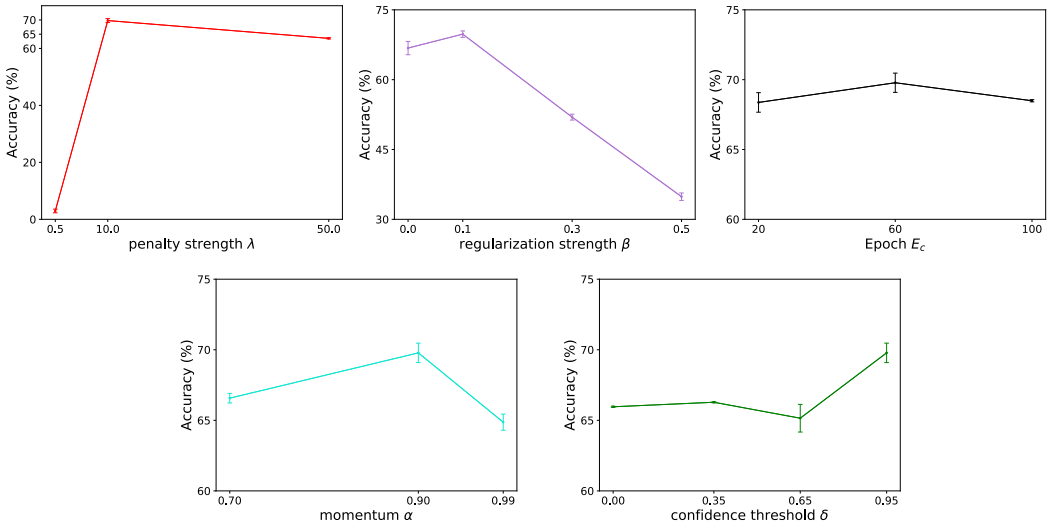


Figure 7: Test accuracy on CIFAR-100 with 60% symmetric label noise. The mean accuracy over three runs is reported, along with bars representing one standard deviation from the mean. In each experiment, the rest of hyperparameters are fixed to the values reported in Section C.4.

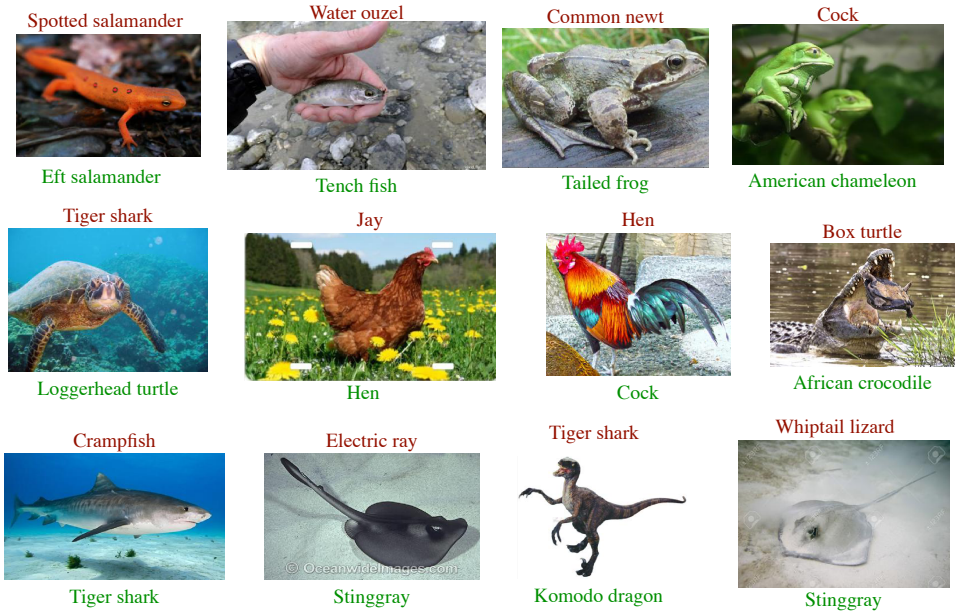


Figure 8: Label correction of Webvision images. Given noisy labels are shown above in red and the corrected labels are shown below in green.

D MORE RESULTS OF LABEL CORRECTION AND CONFIDENCE VALUE

We report the label correction accuracy for various level of label noise on CIFAR-10 and CIFAR-100 in Table 6. Figure 10 displays the confusion matrix of corrected label w.r.t. the clean labels on CIFAR-10 with 60% symmetric, 80% symmetric and 40% asymmetric label noise respectively. We also show the corrected labels for real-world datasets in Figure 8 and Figure 9.

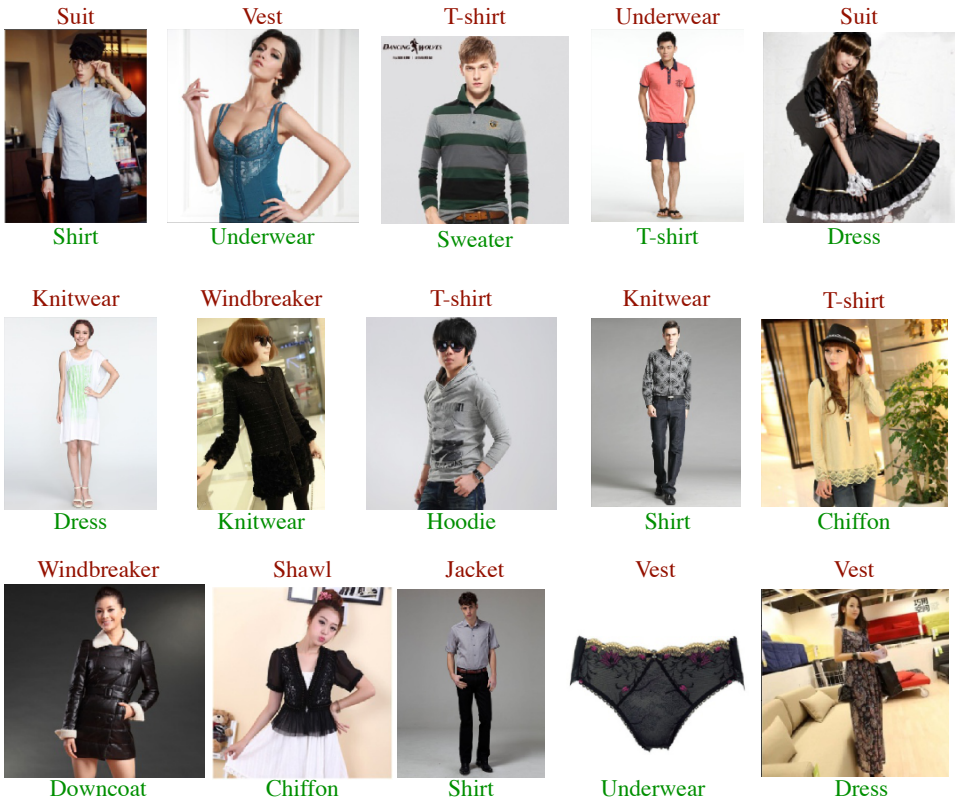


Figure 9: Label correction of Clothing1M images. Given noisy labels are shown above in red and the corrected labels are shown below in green.

We report the confidence value for high level of label noise on CIFAR-10 in Figure 11 and Figure 12. As we can see, the confidence values on the diagonal blocks remain higher than those non-diagonal blocks.

Table 6: Correction accuracy (%) on CIFAR-10 and CIFAR-100 with various levels of label noise injected to training set.

Dataset	CIFAR-10					CIFAR-100				
Noise ratio	symm				asymm	symm				asymm
	20%	40%	60%	80%	40%	20%	40%	60%	80%	40%
Correction accuracy	97.3	95.1	91.1	81.1	93.8	92.6	86.4	76.5	42.4	87.1

E PERFORMANCE WITH DIFFERENT ESTIMATION STRATEGIES

We compare the performance of CAR with three strategies: 1) our strategy in Section 3.4. 2) temporal ensembling (Laine & Aila, 2017) that adopted in ELR (Liu et al., 2020). 3) directly using the noisy labels $\hat{y}$ without label correction. The temporal ensembling calculate the target by

$$\boldsymbol{t}_{[E]}^{[i]} = \begin{cases} \mathbf{0} & \text{if } E < E_c \\ \alpha \boldsymbol{t}_{[E-1]}^{[i]} + (1-\alpha)\boldsymbol{p}_{[E]}^{[i]} & \text{if } E \geq E_c \\ \boldsymbol{t}_{[E-1]}^{[i]} & \text{otherwise,} \end{cases} \tag{22}$$

where the target $\boldsymbol{t}$ solely depends on model prediction. Table 7 shows the results. As we can see, compared to CAR without label correction, CAR with temporal ensembling does not improve much

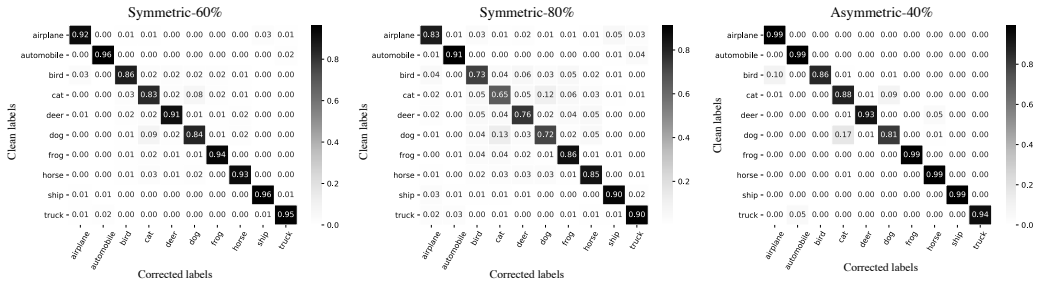


Figure 10: Confusion matrix of corrected labels w.r.t clean labels on CIFAR-10 with 60% symmetric, 80% symmetric and 40% asymmetric label noise respectively.

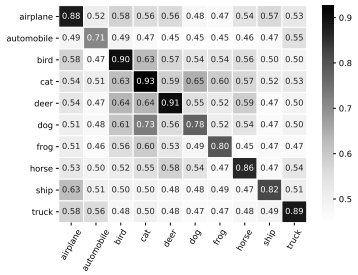


Figure 11: Average confidence values τ of false labels w.r.t clean labels on CIFAR-10 with 60% symmetric label noise.

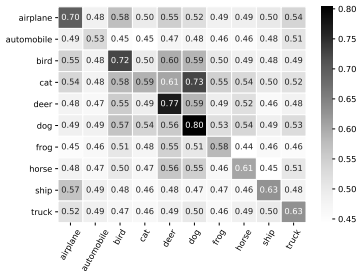


Figure 12: Average confidence values τ of false labels w.r.t clean labels on CIFAR-10 with 80% symmetric label noise.

performance in easy cases (e.g. 40% symmetric label noise), and it even gets worse performance in hard cases (e.g. 80% symmetric label noise). However, CAR with our strategy achieves much better performance. We also conduct the experiments that use CE with different target estimation strategies. Surprisingly, CE with our strategy can achieve better performance to CAR in CIFAR-10 with 40% asymmetric noise. However, the overall performance is worse than the performance of using CAR, due to the reason that CE will memorize noisy labels after early learning phase.

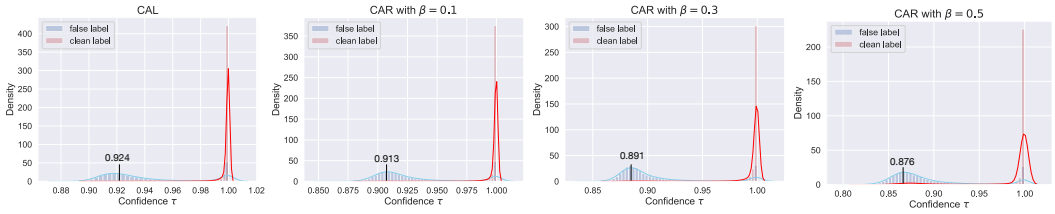


Figure 13: The empirical density of confidence value τ on CIFAR-100 with 40% symmetric label noise. The mean confidence values of mislabeled samples become smaller with the increasing of β.

F INFLUENCE OF $\mathcal{L}_{r\text{-}cace}$ ON CONFIDENCE DISTRIBUTION

The empirical results of the influence of confidence distribution on CIFAR-100 with different strengths of $\mathcal{L}_{r\text{-}cace}$ are shown in Figure 13. We can observe that with the larger coefficient β on $\mathcal{L}_{r\text{-}cace}$, the averge confidence of mislabeled samples is closer to 0. Add the different strengths auxililary term $\mathcal{L}_{r\text{-}cace}$ to CAL does further segregate the mislabeled samples from the clean samples.

Table 7: The test accuracy of CAR and CE with different target estimation strategies. All the following experiments use Cosine Annealing learning rate scheduler (Loshchilov & Hutter, 2017).

Dataset	CIFAR-10			CIFAR-100		
Noise type	symm		asymm	symm		asymm
Noise ratio	40%	80%	40%	40%	80%	40%
CAR with our strategy	**93.49 ± 0.07**	**80.98 ± 0.27**	**92.09 ± 0.12**	**75.38 ± 0.08**	**38.24 ± 0.55**	**74.89 ± 0.20**
CAR with temporal ensembling	89.52 ± 0.30	64.07 ± 2.04	80.52 ± 2.21	70.80 ± 0.38	10.28 ± 1.67	63.91 ± 1.65
CAR w/o label correction	89.47 ± 0.50	76.91 ± 0.22	88.23 ± 0.22	69.91 ± 0.21	31.33 ± 0.38	55.68 ± 0.17
CE with our strategy	92.64 ± 0.21	75.51 ± 0.38	**92.21 ± 0.11**	68.53 ± 0.47	32.36 ± 0.44	73.01 ± 0.90
CE with temporal ensembling	92.12 ± 0.16	72.87 ± 1.98	89.71 ± 1.43	70.45 ± 0.22	9.34 ± 0.78	66.38 ± 0.57
CE w/o label correction	78.26 ± 0.74	56.42 ± 2.49	86.55 ± 1.06	46.34 ± 0.56	11.55 ± 0.35	48.86 ± 0.04

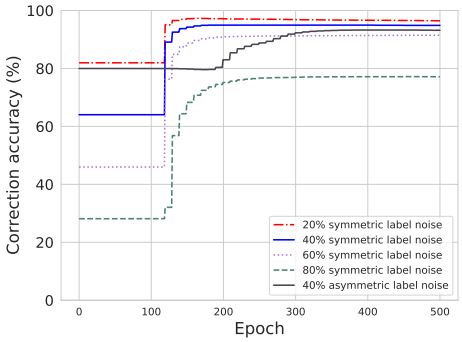


Figure 14: Label correction accuracy vs. epochs on CIFAR-10 with different levels of label noise.

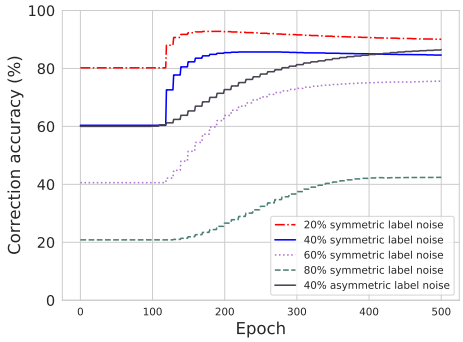


Figure 15: Label correction accuracy vs. epochs on CIFAR-100 with different levels of label noise.

G STABILITY OF ITERATIVE LABEL CORRECTION

We plot the CIFAR-10 and CIFAR-100 label correction accuracy vs. epochs in Figure 14 and Figure 15 respectively. In both datasets, our iterative label correction strategy achieves a stable correction effect and the correction accuracy does not drop with the increasing of training epochs. In summary, iterative label correction does not only recovers the noisy labels back to clean labels but also achieves high correction accuracy. In addition, the correction accuracy remains stable, which demonstates that incorporating a certain percentage of prediction to update the noisy labels is an efficient and reliable way to correct noisy labels.