A Gift from Label Smoothing: Robust Training with Adaptive Label Smoothing via Auxiliary Classifier under Label Noise

Jongwoo Ko[1*], **Bongsoo Yi**[2*], **Se-Young Yun**[1]

[1] Kim Jaechul Graduate School of AI, KAIST, Seoul, Korea
[2] Department of Statistics and Operations Research, University of North Carolina at Chapel Hill
{jongwoo.ko, yunseyoung}@kaist.ac.kr, bongsoo@unc.edu

Abstract

As deep neural networks can easily overfit noisy labels, robust training in the presence of noisy labels is becoming an important challenge in modern deep learning. While existing methods address this problem in various directions, they still produce unpredictable sub-optimal results since they rely on the posterior information estimated by the feature extractor corrupted by noisy labels. Lipschitz regularization successfully alleviates this problem by training a robust feature extractor, but it requires longer training time and expensive computations. Motivated by this, we propose a simple yet effective method, called ALASCA, which efficiently provides a robust feature extractor under label noise. ALASCA integrates two key ingredients: (1) adaptive label smoothing based on our theoretical analysis that label smoothing implicitly induces Lipschitz regularization, and (2) auxiliary classifiers that enable practical application of intermediate Lipschitz regularization with negligible computations. We conduct wide-ranging experiments for ALASCA and combine our proposed method with previous noise-robust methods on several synthetic and real-world datasets. Experimental results show that our framework consistently improves the robustness of feature extractors and the performance of existing baselines with efficiency.

1 Introduction

While deep neural networks (DNNs) have high expressive power that leads to promising performances, the success of DNNs heavily relies on the quality of training data, in particular, accurately labeled training examples. Unfortunately, labeling large-scale datasets is a costly and error-prone process, and even high-quality datasets contain incorrect labels (Nettleton, Orriols-Puig, and Fornells 2010; Zhang et al. 2017). Hence, mitigating the negative impact of noisy labels is critical, and many approaches have been proposed to improve robustness against noisy data for learning with noisy labels (LNL).

Robustness to label noise is typically pursued by identifying noisy samples to reduce their contribution to the loss (Han et al. 2018; Mirzasoleiman, Cao, and Leskovec 2020), correcting labels (Yi and Wu 2019; Li, Socher, and Hoi 2020), utilizing a robust loss function (Zhang and Sabuncu 2018; Wang et al. 2019). However, one of the biggest challenges of LNL methods involves providing a dependable criterion for distinguishing clean data from noisy data, such that clean data is fully exploited while filtering noisy data. While these existing methods are partially effective in mitigating label noise, their criterion for identifying noisy examples uses biased posterior information from a linear classifier or the penultimate layer of the corrupted network. These unpredictable biases can lead to a reduction in the network's ability to separate clean and noisy instances (Nguyen et al. 2020; Kim et al. 2021a).

To solve this undesired bias, several regularization methods (Xia et al. 2021; Cao et al. 2021) have been proposed to enhance the robustness of the feature extractor. However, while existing regularization-based learning frameworks alleviate the degradation, these methods require multiple training stages and considerable computational costs and are difficult to apply in practice. Cao et al. (2021) used two-stage training to compute the relative data-dependent regularization power to conduct Lipschitz regularization (LR) on intermediate layers. Xia et al. (2021) identified and regularized the non-critical parameters that tend to fit noisy labels and require longer training time. Some studies (Zhang and Yao 2020; Zheltonozhskii et al. 2022) have designed contrastive learning frameworks to generate high-quality feature extractors using unsupervised approaches, which require considerable computations for high performance.

To mitigate these impractical issues, we provide a simple yet effective learning framework for a robust feature extractor, **A**daptive **LA**bel **S**moothing via auxiliary **Cl**A**ssifier (ALASCA), with theoretical guarantee and small additional computation. Our proposed method is robust to label noise itself and can further enhance the performance of existing LNL methods. Our main contributions are as follows:

- We theoretically explain that label smoothing (LS) implicitly induces LR, which is known to enable robust training with noisy labels (Finlay et al. 2018; Cao et al. 2021). Through theoretical motivations, we empirically show that adaptive LS (ALS) can regularize noisy examples while fully exploiting clean examples.
- To practically implement adaptive LR on the intermediate layers, we propose ALASCA, which combines ALS with auxiliary classifiers. To the best of our knowledge, this is the first study to apply auxiliary classifiers under

[*]The two authors contributed equally.

label noise with theoretical evidence.

- We experimentally demonstrate that ALASCA is universal by combining various LNL methods and validating that ALASCA consistently boosts robustness on benchmark-simulated and real-world datasets.
- We verify that ALASCA effectively enhances the robustness of feature extractors by comparing the quality of subsets on sample-selection methods and robustness to the hyperparameter selection of LNL methods.

2 Related Works

2.1 Learning with Noisy Labels

Zhang et al. (2017) empirically demonstrated that convolutional neural networks trained with stochastic gradient methods easily memorize random labeling of the training data. To address this, numerous studies have examined the classification task with noisy labels. Existing methods address this problem by (1) filtering noisy examples and training using only clean examples (Han et al. 2018; Mirzasoleiman, Cao, and Leskovec 2020; Kim et al. 2021a) or (2) relabeling noisy examples using the model itself or another model trained only on the clean dataset (Lee et al. 2018; Li, Socher, and Hoi 2020). Some approaches focus on designing loss functions with robust behaviors and provable tolerance to label noise (Ghosh, Kumar, and Sastry 2017; Zhang and Sabuncu 2018; Wang et al. 2019). We fully describe these previous works in Appendix B.

Regularization-based Methods. Another line of work has attempted to design regularization-based techniques. For example, some studies have stated and theoretically analyzed how early-stopped model can prevent the memorization phenomenon of noisy labels (Arpit et al. 2017; Song et al. 2019). Based on this, Liu et al. (2020) proposed an early learning regularization (ELR) loss function that avoids memorizing noisy data by leveraging semi-supervised learning (SSL) techniques. Xia et al. (2021) clarified that neural network parameters cause memorization and proposed a robust training method for these parameters. Developing regularization at the prediction level has been addressed by smoothing one-hot vectors (Lukasik et al. 2020) and distilling the rescaled predictions of other models (Müller, Kornblith, and Hinton 2019; Kim et al. 2021b). Recently, Cao et al. (2021) proposed a heteroskedastic adaptive regularization that applies stronger regularization to noisy instances.

2.2 Label Smoothing

LS (Szegedy et al. 2016; Müller, Kornblith, and Hinton 2019) is commonly used to construct a generalized DNN model by preventing over-confident predictions. This regularization technique facilitates generalization by softening a ground-truth one-hot vector $\mathbf{y}$ with a weighted mixture of hard targets:

$$\mathbf{y}^{LS} := (1-\alpha)\cdot\mathbf{y} + \frac{\alpha}{L}\cdot\mathbf{1}_L,$$

where L denotes the number of classes, $\mathbf{1}_L$ denotes an all-one vector in $\mathbb{R}^L$, and $\alpha \in [0,1]$ is a smoothing parameter. Lukasik et al. (2020) claimed that LS denoises label noise by causing label correction and weight-shrinkage regularization effects. However, Wei et al. (2021a) recently show that LS tends to over-smooth the estimated posterior under high levels of label noise, which can hurt robustness. Moreover, several studies (Szegedy et al. 2016; Pereyra et al. 2017; Müller, Kornblith, and Hinton 2019; Chorowski and Jaitly 2017) have validated that LS boosts model generalizability, and Li, Dasarathy, and Berisha (2020) proposed the need for data-dependent smoothing. Li, Dasarathy, and Berisha (2020) proposed structural LS, which selects smoothing strength data-dependently that minimizes the Bayes error rate bias. Ghoshal et al. (2021) derived PAC Bayesian generalization bounds for LS and proposed adaptive smoothing for the latent structure of the label space.

3 Methodology: ALASCA

LR has been shown to be effective for DNNs (Gouk et al. 2021). Wei and Ma (2019a,b) theoretically and empirically show that LR for all intermediate layers improves generalization of DNNs. Furthermore, many studies show that different regularization strengths along the data points are essential. Wang, Du, and Shen (2013) and Tibshirani (2014) stated that smoothing splines with different smoothing parameters perform well in regression problems. Recently, Cao et al. (2021) showed that applying strong LR to highly uncertain data points improves generalization. We recall key takeaways from Cao et al. (2021) that motivated our work.

Remark 3.1 (Cao et al. 2021)**.** *In a binary classification problem on one-dimensional data, the authors i) derived the formula for the asymptotic mean squared error (MSE) on the test set, ii) with some simplifications, showed that the asymptotic MSE is minimized when the smoothing parameter is proportional to the $\frac{3}{5}$-th power of the label uncertainty.*

The exact theorem statement and detailed explanation may be found in Appendix A.1. However, this explicit regularization requires multiple training phases to estimate and apply relative regularization power for different data points. Furthermore, computing the Hessian matrix resulting from directly regularizing the norm of Jacobian matrices increases the computational cost (Filiposka, Djuric, and ElMaraghy 2014; Nesser et al. 2021). In this section, we present that LS implicitly incurs LR and introduce the simple unified framework for efficient learning with label noise, ALASCA. In principle, our method can be used with most LNL methods, such as noise-robust loss function (Zhang and Sabuncu 2018; Wang et al. 2019) and sample-selection methods (Han et al. 2018; Kim et al. 2021a).

3.1 Label Smoothing as Lipschitz Regularization

In this section, we analytically present our motivation that LS implicitly encourages LR. Here, we formally define the notation and terminology for our problem.

Notation. We focus on multiclass classification with L classes. Assume that the data points and labels lie in $\mathcal{X}\times\mathcal{Y}$, where the feature space $\mathcal{X} \subset \mathbb{R}^D$ and label space $\mathcal{Y} =$

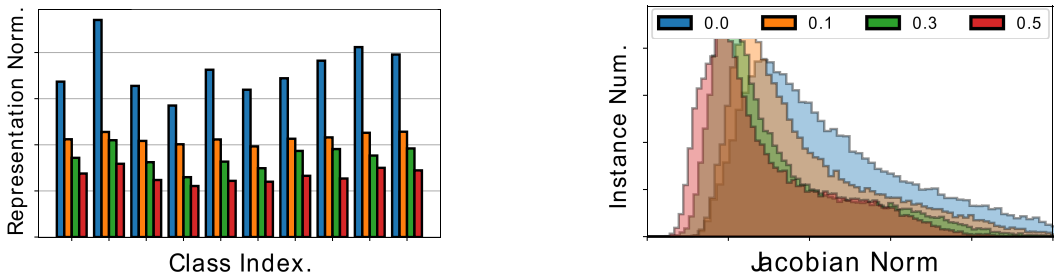


Figure 1: Comparison of class-wise representation vector norm (left; Theorem 3.5), and distribution of Jacobian matrix norm for penultimate layer (right; Theorem 3.7) across different smoothing factors on CIFAR-10.

$\{0,1\}^L$. A single data point $\mathbf{x}$ and its label $\mathbf{y}$ follow a distribution $(\mathbf{x},\mathbf{y}) \sim P_{\mathcal{X}\times\mathcal{Y}}$. We aim to find a predictor $\mathbf{f} : \mathcal{X} \to \mathbb{R}^L$ minimizing the risk of $\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim P_{\mathcal{X}\times\mathcal{Y}}}[\ell(\mathbf{f}(\mathbf{x}),\mathbf{y})]$ with loss function $\ell : \mathbb{R}^L \times \mathbb{R}^L \to \mathbb{R}_+$.

Definition 3.2 (Lipschitzness)**.** A function f is called Lipschitz continuous with a Lipschitz constant $L_f \in [0,\infty)$ if

$$\|f(y) - f(x)\| \leq L_f \|y - x\|,$$

for all $x, y \in \text{dom } f$.

Definition 3.3 (Lipschitz Regularization)**.** Let $\mathcal{F}$ be a twice-differentiable model family from $\mathbb{R}^D$ to $\mathbb{R}^L$. Lipschitz regularization aims to optimize the function with a smoothness penalty as follows:

$$\min_{\mathbf{f}\in\mathcal{F}} \frac{1}{N}\sum_{n=1}^{N} \ell(\mathbf{f}(\mathbf{x}_n),\mathbf{y}_n) + \lambda \|\mathbf{J}_\mathbf{f}(\mathbf{x}_n)\|_F,$$

where λ is the regularization coefficient, N the number of training data points, $\mathbf{J}_\mathbf{f}$ the Jacobian matrix of $\mathbf{f}$, and $\|\cdot\|_F$ the Frobenius norm.

Here, we focus only on ℓ as cross-entropy (CE) loss. Compared with the CE loss with one-hot vector $\ell(\mathbf{f}(\mathbf{x}),\mathbf{y})$, the CE loss with LS $\ell(\mathbf{f}(\mathbf{x}),\mathbf{y}^{LS})$ of factor α can be presented as follows:

$$\begin{aligned}\ell(\mathbf{f}(\mathbf{x}),\mathbf{y}^{LS}) &= (1-\alpha)\cdot\ell(\mathbf{f}(\mathbf{x}),\mathbf{y}) + \frac{\alpha}{L}\cdot\ell(\mathbf{f}(\mathbf{x}),\mathbf{1}_L) \\ &\propto \ell(\mathbf{f}(\mathbf{x}),\mathbf{y}) + \frac{\alpha}{(1-\alpha)\cdot L}\cdot\Omega(\mathbf{f}).\end{aligned}$$

By denoting $f_i(\cdot)$ as the i-th element of logit vector $\mathbf{f}(\cdot)$, the regularization term of LS is

$$\Omega(\mathbf{f}) = L\cdot\log\left[\sum_{i=1}^{L} e^{f_i(\cdot)}\right] - \sum_{i=1}^{L} f_i(\cdot). \tag{1}$$

Previously, Lukasik et al. (2020) suggested that LS encourages weight shrinkage in DNNs; however, this interpretation was validated only for linear models. To better understand LS, we consider that the DNN function can be presented as arbitrary surrogate models. We suppose that the surrogate model $\mathbf{f}(\cdot) = \mathbf{g}\circ\mathbf{h}(\cdot)$ consists of an arbitrary twice-differentiable feature extractor $\mathbf{h} : \mathcal{X} \to \mathbb{R}^Q$ and fixed linear classifier $\mathbf{g}(\mathbf{z}) = \mathbf{W}^\intercal\mathbf{z}$ with $\mathbf{W} \in \mathbb{R}^{Q\times L}$ and $\mathbf{z} \in \mathbb{R}^Q$. First, we find a minimizer of the regularization term of LS. The following assumption is required to guarantee the uniqueness of the minimizer.

Assumption 3.4. $\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_L$ is an affine basis of $\mathbb{R}^Q$, where $\mathbf{W}_i$ is the i-th column of $\mathbf{W}$ (i.e., $\mathbf{W}_2 - \mathbf{W}_1, \mathbf{W}_3 - \mathbf{W}_1, \cdots, \mathbf{W}_L - \mathbf{W}_1$ are linearly independent).

Theorem 3.5. $\mathbf{h} = \mathbf{0}$ *is a minimizer of* $\Omega\circ\mathbf{g}$. *If Assumption 3.4 holds,* $\mathbf{h} = \mathbf{0}$ *is the unique minimizer.*

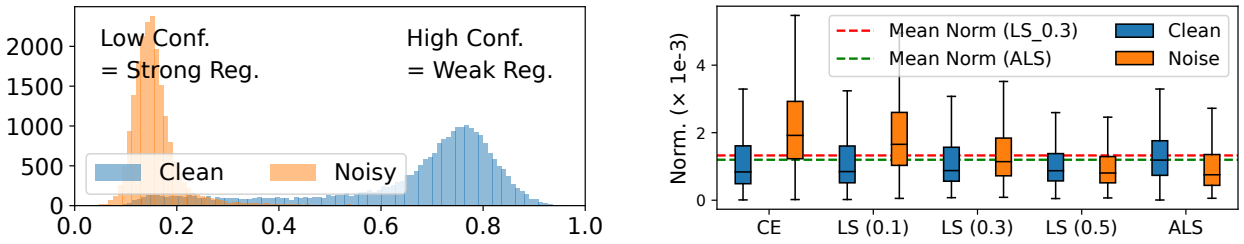
Note that $\mathbf{h} = \mathbf{0}$ is always a minimizer of Equation (1) without any assumptions. The takeaway from Theorem 3.5 is that the regularization term of LS encourages $\mathbf{h}(\mathbf{x}_1),\ldots,\mathbf{h}(\mathbf{x}_N)$ to shrink to zero. However, this does not ensure the shrinkage of the Jacobian matrix of $\mathbf{f}$. The following theorem shows that this is true under a common Lipschitz assumption. We state the assumption and present our next main result.

Assumption 3.6. Each gradient $\nabla h_i(\mathbf{x})$ is Lipschitz continuous with a Lipschitz constant L_h for all i, where $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), h_2(\mathbf{x}), \cdots, h_Q(\mathbf{x}))$.

Theorem 3.7. *Consider a bounded feature space* $\mathcal{X}$ *and suppose that Assumption 3.6 is satisfied. If* $\mathbf{h}(\mathbf{x}_n) = \mathbf{0}$ *for* $1 \leq n \leq N$, $\|\mathbf{J}_\mathbf{f}(\mathbf{x}_n)\|_F \to \mathbf{0}$ *as* $N \to \infty$ *holds for* $1 \leq n \leq N$.

Theorem 3.7 states that the smoothness of the feature extractor function in the training set induces LR. By combining Theorem 3.5 and 3.7, we conclude that the regularization term of LS encourages LR. The detailed proof of the theorems may be found in Appendix A.2.To validate our theoretical perspective, we conducted the following exploratory experiments: compare the (1) class-wise average norm values of representation vectors and (2) Jacobian matrix norm from the penultimate layer of ResNet34 trained on CIFAR-10. As Figure 1 shows, both the representation and Jacobian matrix norms decrease to zero as the smoothing factor increases, indicating that the LS regularization term induces the representation vector to the origin (Theorem 3.5) and implicitly incurs LR (Theorem 3.7).

Adaptive Regularization. From our perspective of LS, we can apply adaptive LR using different smoothing factors. Cao et al. (2021) shows that applying stronger LR to highly uncertain data points improves generalization on noisy datasets. To implement adaptive LR through LS, we design the smoothing factor of LS proportional to the $1 - \Pr(y|\mathbf{x})$ for each instance. While Cao et al. (2021) suggests that the optimal smoothing parameter is proportional to the



(a) Concept of ALS

(b) Jacobian Norm

Figure 2: (a) The concept of ALS. Because noisy instances tend to have lower confidence, we conduct stronger regularization on lower confidence instances. (b) Comparison of Jacobian matrix norms for the penultimate representation on CIFAR-10 with 50% of symmetric noise.

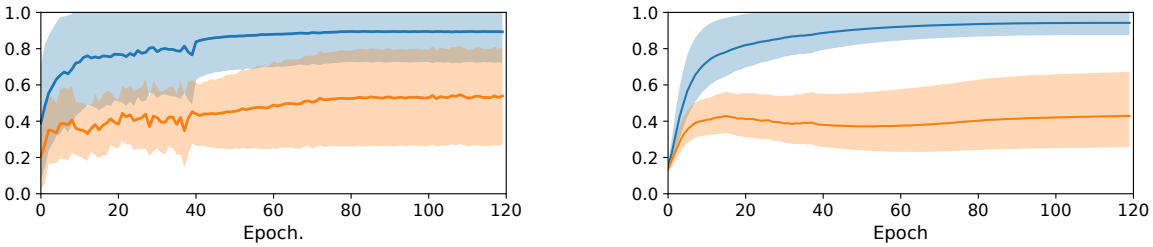


Figure 3: Dynamic patterns (mean $\pm$ std) of instantaneous (left) and EMA confidence (right) suggested in ALASCA for CIFAR-10 under 40% of asymmetric noise. The blue and orange lines are corresponding to clean and noisy instances.

3/5-th power of the label uncertainty, our proposed strategy shows similar performances despite its simplicity. Consequently, we use the following loss function for ALS:

$$\ell^{ALS}_{\tilde{\alpha}(\mathbf{x})}(\mathbf{f}(\mathbf{x}),\mathbf{y}) = (1-\tilde{\alpha}(\mathbf{x}))\cdot\ell(\mathbf{f}(\mathbf{x}),\mathbf{y}) + \tilde{\alpha}(\mathbf{x})\cdot\Omega(\mathbf{f}), \quad (2)$$

where $\tilde{\alpha}(\mathbf{x})$ is the y-th element of $1-\mathcal{S}(\mathbf{f}(\mathbf{x}))$, and $\mathcal{S}$ is the softmax function. Figure 2 shows that the proposed ALS enables us to mainly regularize noisy examples because we conduct stronger regularization for lower confidence examples, where noisy examples take up a large proportion. While uniform LS (ULS) equally regularizes the smoothness of both clean and noisy examples, we validate that ALS can apply the appropriate LR for each example.

3.2 Combination with Additional Techniques

Now, we integrate ALS with auxiliary classifiers (AC) and the exponential moving averaged (EMA) confidence to practically regularize the smoothness of intermediate layers. We describe the overall algorithm of ALASCA in Algorithm 1.

Use of Auxiliary Classifiers. Recent studies have emphasized LR of intermediate layers. Wei and Ma (2019a,b) showed that LR for all intermediate layers improves generalization of DNNs. Sokolić et al. (2017) and Cao et al. (2021) showed similar results in data-limited and distribution shift setups, respectively. Inspired by these works, we use LS as an intermediate LR by utilizing an AC commonly used in deep learning. While ACs work well on various domains such as self-distillation (Zhang et al. 2019) and class-imbalance (Lee, Shin, and Kim 2021), our work first proposes using ACs with noisy labels based on theoretical motivation. We compare our method with Zhang et al. (2019) in D.2.We apply bottleneck (Howard et al. 2017) architecture as ACs and attach them at the end of all blocks of backbone by our results in Section 4.3. By applying ACs, we can encourage greater LR to noisy instances and weaker LR to clean instances with only a small additional computation. Furthermore, this approach has two additional advantages. Wei et al. (2021b) showed that the benefits of LS disappear in a high-level noise, as applying LS tends to over-smooth

Algorithm 1: ALASCA

Require: $\{\mathbf{x}_i, \mathbf{y}_i\}, 1 \le i \le N$
Require: $\mathcal{L}$ ▷ Loss function for existing LNL methods
Require: $\{\Theta_k\}_{k=0}^{K}$ ▷ Parameters for main and ACs
Require: β, τ ▷ EMA weight and temperature of ALASCA
Require: λ ▷ Coefficient for power of regularization
Output: Θ_0

$\mathbf{t}_i \leftarrow \mathbf{0}_{\mathrm{N}}$. ▷ Initialize EMA confidence
for each minibatch B **do**
  $\mathbf{t}_i \leftarrow \beta\mathbf{t}_i + (1-\beta)\mathbf{f}_{\Theta_0}(\mathbf{x}_i)$ ▷ EMA (Averaging)
  $\tilde{\alpha}(\mathbf{x}_i) \leftarrow 1 - \mathcal{S}(\mathbf{t}_i/\tau)$ ▷ EMA (Sharpening)
  loss $\leftarrow -\frac{1}{|B|}\sum_{i=1}^{|B|}\mathcal{L}\left(\mathbf{f}_{\Theta_0}(\mathbf{x}_i), \mathbf{y}_i\right)$
  $+\frac{\lambda}{|B|}\sum_{k=1}^{K}\sum_{i=1}^{|B|}\ell^{ALS}_{\tilde{\alpha}(\mathbf{x}_i)}\left(\mathbf{f}_{\Theta_k}(\mathbf{x}_i), \mathbf{y}_i\right)$
  /* Compute loss by using Eq. (2) */
  update Θ_0 and $\{\Theta_k\}_{k=1}^{K}$ using SGD
end for

Dataset	CIFAR-10				CIFAR-100			
Noisy Type	Symm.		Asym.	Inst.	Symm.			Asym.
Noise Ratio	50%	80%	40%	40%	20%	50%	80%	40%
Standard	78.2 ± 0.8	53.8 ± 1.0	85.0 ± 0.1	74.1 ± 2.9	58.7 ± 0.3	42.5 ± 0.3	18.1 ± 0.8	42.7 ± 0.6
+ ALASCA	$\mathbf{88.0 \pm 0.3}$	$\mathbf{70.3 \pm 0.4}$	$\mathbf{90.3 \pm 0.3}$	$\mathbf{81.4 \pm 0.3}$	$\mathbf{70.6 \pm 0.2}$	$\mathbf{59.7 \pm 0.4}$	$\mathbf{26.1 \pm 0.9}$	$\mathbf{59.0 \pm 0.6}$
GCE	86.5 ± 0.2	64.1 ± 1.4	76.7 ± 0.6	55.7 ± 0.2	66.8 ± 0.4	57.3 ± 0.3	29.2 ± 0.7	47.2 ± 1.2
+ ALASCA	$\mathbf{88.4 \pm 0.3}$	$\mathbf{73.7 \pm 1.5}$	$\mathbf{90.2 \pm 0.1}$	$\mathbf{73.5 \pm 1.2}$	$\mathbf{70.8 \pm 0.5}$	$\mathbf{60.1 \pm 0.4}$	$\mathbf{32.2 \pm 0.5}$	$\mathbf{57.3 \pm 0.5}$
SCE	84.7 ± 0.3	68.1 ± 0.8	82.5 ± 0.5	71.4 ± 1.3	70.4 ± 0.1	48.8 ± 1.3	25.9 ± 0.4	48.4 ± 0.9
+ ALASCA	$\mathbf{88.5 \pm 0.1}$	$\mathbf{71.7 \pm 0.8}$	$\mathbf{89.4 \pm 0.2}$	$\mathbf{78.0 \pm 0.6}$	$\mathbf{71.4 \pm 0.2}$	$\mathbf{61.3 \pm 0.6}$	$\mathbf{28.7 \pm 0.6}$	$\mathbf{57.3 \pm 0.7}$
ELR	88.2 ± 0.1	72.9 ± 0.6	90.1 ± 0.5	79.8 ± 0.2	74.2 ± 0.2	59.1 ± 0.8	29.8 ± 0.6	73.3 ± 0.4
+ ALASCA	$\mathbf{89.5 \pm 0.3}$	$\mathbf{74.2 \pm 1.2}$	$\mathbf{90.4 \pm 0.2}$	$\mathbf{82.2 \pm 0.5}$	$\mathbf{74.8 \pm 0.1}$	$\mathbf{63.6 \pm 0.6}$	$\mathbf{34.4 \pm 0.4}$	$\mathbf{73.9 \pm 0.5}$
Co-teaching	83.3 ± 0.6	66.3 ± 1.5	88.4 ± 2.8	70.5 ± 0.5	63.4 ± 0.0	49.1 ± 0.4	20.5 ± 1.3	47.7 ± 1.2
+ ALASCA	$\mathbf{90.1 \pm 1.5}$	$\mathbf{71.1 \pm 1.2}$	$\mathbf{91.2 \pm 0.1}$	$\mathbf{76.8 \pm 0.4}$	$\mathbf{75.5 \pm 0.3}$	$\mathbf{68.1 \pm 0.4}$	$\mathbf{42.2 \pm 1.2}$	$\mathbf{64.7 \pm 0.4}$
CRUST	87.0 ± 0.1	64.8 ± 1.1	82.4 ± 0.0	64.7 ± 2.1	69.3 ± 0.2	62.3 ± 0.2	21.7 ± 0.7	56.1 ± 0.5
+ ALASCA	$\mathbf{87.6 \pm 0.2}$	$\mathbf{71.5 \pm 1.5}$	$\mathbf{90.2 \pm 0.2}$	$\mathbf{71.6 \pm 1.1}$	$\mathbf{70.4 \pm 0.1}$	$\mathbf{64.1 \pm 0.9}$	$\mathbf{25.5 \pm 0.7}$	$\mathbf{58.3 \pm 0.5}$
FINE	87.3 ± 0.2	69.4 ± 1.1	89.5 ± 0.1	82.4 ± 0.5	70.3 ± 0.2	64.2 ± 0.5	25.6 ± 1.2	61.7 ± 1.0
+ ALASCA	$\mathbf{88.0 \pm 0.1}$	$\mathbf{70.6 \pm 0.9}$	$\mathbf{90.3 \pm 0.2}$	$\mathbf{84.3 \pm 1.2}$	$\mathbf{70.9 \pm 0.3}$	$\mathbf{65.8 \pm 0.2}$	$\mathbf{29.4 \pm 1.5}$	$\mathbf{63.5 \pm 0.7}$

Table 1: Test accuracies (%) on CIFAR-10/-100 under different noise types and fractions for noise-robust loss and sample-selection approaches. The results for symmetric and asymmetric noise of all baseline methods were taken from Kim et al. (2021a). Instance-dependent noise results are reported by our re-implementation based on official codes. The average accuracies over three trials are reported. The best results sharing the noisy fraction and method are highlighted in bold.

the estimated posterior of the main classifier. However, LS with ACs does not affect the main classifier but effectively regularizes the Lipschitzness of intermediate layers. Moreover, as we use multiple classifiers, we obtain more robust predictions from the ensemble effect during inference. In Algorithm 1, we denote the parameters of the main classifier and ACs as Θ_0 and $\{\Theta_k\}_{k=1}^{K}$, where K is number of AC. We further denote outputs of the k-th classifier as $\mathbf{f}_{\Theta_k}(\cdot)$.

Use of EMA Confidence. As shown in Figure 3, instantaneous confidence suffers from high variance across training epochs and is inaccurate for differentiating regularization power between clean and noisy instances. Incorrect regularization power caused by such instability leads to performance degradation. Hence, we use EMA along the epochs to compute confidence and effectively obtain the appropriate regularization power. In SSL, this weight averaging approach has been proposed to mitigate confirmation bias (Liu et al. 2020). The computation procedure of EMA confidence is as follows. (1) To reduce variance and enhance stability, we conduct EMA on output values (Line 3 in Algorithm 1). (2) Because the averaged outputs are over-smooth, which causes weak regularization for noisy examples and strong regularization for clean examples, we sharpen the EMA logits by dividing sharpen temperature τ (Line 4 in Algorithm 1). We observe that regularization powers on clean and noisy examples are clearly distinguished and become stable after using EMA confidence, as shown in Figure 3.

4 Experiments

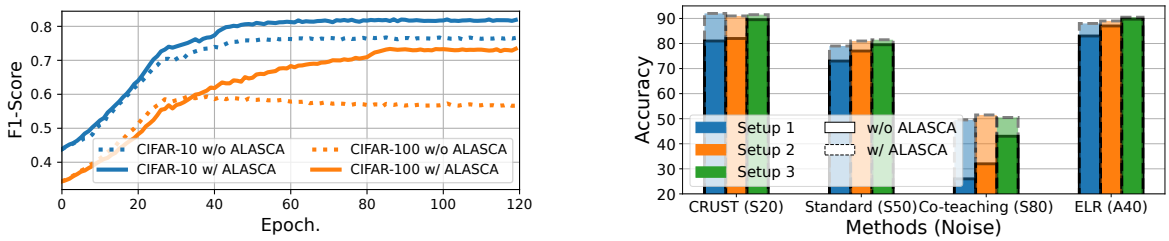
We design experiments to answer the following questions:

- Can ALASCA improve existing LNL methods, such as noise-robust loss functions and sample-selection methods for both synthetic and real-world datasets? (Section 4.1 & 4.4)
- How effective is ALACSA in improving the robustness of the feature extractor? (Section 4.2)
- How do the architecture and position of ACs affect the performance and efficiency? & Which component is important to the performance in ALASCA? (Section 4.3)

4.1 Experimental Setup and Results on CIFAR

Setup. We inject uniform randomness into a fraction of labels for symmetric noise and flip labels to specific classes for asymmetric noise by following Kim et al. (2021a). To set up instance-dependent noise, we follow the noise generation of Cheng et al. (2020). We use the architectures of backbone network and hyperparameter settings for all baseline experiments following Kim et al. (2021a). We set β, τ, and λ as 0.7, 1/3, and 2.0, respectively. The detailed experimental setup is described in Appendix C.1. and Appendix C.2.To verify the superiority of our method, we combine ALASCA with various existing LNL methods (noise-robust loss functions and sample-selection methods) and identify that ours consistently improves the generalization in the presence of noisy data. Furthermore, we perform additional experiments incorporating semi-supervised approaches with ALASCA. Appendix D.1 provides detailed description and results for the SSL approaches.

Noise-Robust Loss Functions. Noise-robust loss functions aim to achieve high performances for unseen clean data



(a) Co-teaching (F1-Score)

(b) Hyperparameter Selection

Figure 4: (a) Comparison of F1-scores of Co-teaching with and without ALASCA on CIFAR-10 and CIFAR-100 under 80% of symmetric noise. (b) Comparison of test accuracies (%) along different hyperparameter settings for various LNL methods. Through the results, we verify that ALASCA enhances the robustness of feature extractors.

despite the presence of noisy labels in the training data. We combine our proposed method with three loss functions: (1) standard CE (Standard); (2) generalized CE (GCE; Zhang and Sabuncu 2018), which can be seen as a generalization of the mean absolute error and standard CE; (3) symmetric CE (SCE; Wang et al. 2019), which is the weighted sum of CE and reverse version of CE; and (4) early learning regularization (ELR; Liu et al. 2020) which uses a regularization term that incorporates target probabilities from the model output. We observe that ALASCA improves generalization when applied to the noise-robust loss function of Table 1.

Sample-Selection Methods. Sample-selection methods, which select clean sample candidates from the training dataset, are a popular direction in LNL. We combine ALASCA with the following sample-selection approaches: (1) Co-teaching (Han et al. 2018), which utilizes two networks, extracts subsets of examples with small losses from each network, and trains each network with subsets of examples filtered by another network; (2) CRUST (Mirzasoleiman, Cao, and Leskovec 2020), which selects a subset of small weight gradient instances; and (3) FINE (Kim et al. 2021a), which selects instances whose penultimate vector highly correlates with the class-representative eigenvector. Table 1 shows the performance increase of ALASCA with different sample-selection methods on various label noise.

4.2 Quality of Feature Extractors

Many LNL methods use posterior information with undesired bias from the corrupted networks under noisy labels, which can lead to sub-optimal performances. However, if the feature extractor is robustly trained under label noise, we can employ unbiased posterior information and achieve better generalization. To validate the effectiveness of ALASCA in terms of improving the robustness of the feature extractor and mitigating undesirable biases, we conduct exploratory experiments: (1) comparison of quality for subsets of sample-selection approaches and (2) robustness to hyperparameter selection of existing LNL methods.

Quality of Sample Selection. To verify that our proposed method robustly trains the feature extractor on label noise, we compute the F1-score for all training epochs to evaluate noise sample filtering in sample-selection methods for various symmetric and asymmetric label noise. We compare the quality of sample selection with and without ALASCA on the Co-teaching. In Figure 4, the F1-scores of sample selection with ALASCA are consistently higher on various label noise than the baseline. Although baseline approaches use different criteria to filter noisy instances (Co-teaching with loss values), we observe that ALASCA improves the quality of all subsets and is effective in training robust feature extractors.

Robust to Hyperparameter Selection. Existing LNL methods have large performance differences depending on their hyperparameters, and, if the hyperparameter value is improperly selected, the performance is provably lower than that of standard training. However, the value of the optimal hyperparameters depends on the network architecture and dataset. We combine ALASCA and existing LNL methods with various hyperparameter settings: (1) Standard with different weight decay factors; (2) ELR with different regularization coefficients; (3) Co-teaching along different warmup epochs; and (4) CRUST with different coreset sizes. The detailed experiment setup and results are in Appendix C.3 and Figure 4b, respectively. While the baseline performances vary depending on the hyperparameters, performances with ALASCA are robust even with different hyperparameters.

4.3 Ablation Studies

To obtain further intuition on ALASCA, we conduct an ablation study on each component of our method. We design three experiments: (1) compare existing methods in terms of performance and efficiency; (2) investigate performance tendency along the structure of the AC; and (3) component analysis to understand the influence of each component.

Efficiency of ALASCA. We compare our implicit regularization framework with the following regularization-based approaches: (1) HAR (Cao et al. 2021), which explicitly and adaptively regularizes the Jacobian matrix norm of data points in higher-uncertainty, lower-density regions more heavily and (2) CDR (Xia et al. 2021), which identifies and regularizes non-critical parameters that tend to fit noisy labels and cannot generalize well. These methods are similar to our proposed method for regularizing the intermediate layer, but are explicit regularization methods that

Method	Standard	GCE	SCE	Co-teaching	FINE	DivideMix	ELR+	ALASCA	A-Coteaching	A-ELR+
Accuracy	68.94	69.75	71.02	70.15	72.91	74.76	74.81	73.78	74.20	**74.92**

Table 2: Comparison of test accuracy (%) on Clothing1M dataset. Results for baselines are obtained from Liu et al. (2020) and Kim et al. (2021a). A-Coteaching and A-ELR+ denote the methods combining ALASCA with Co-teaching and ELR+.

Dataset	CIFAR-10		Efficiency	
Noisy Type	Symm.	Inst.	Mem.	Time
Standard	81.9 ± 0.3	74.1 ± 2.9	$\times$ 1.0	$\times$ 1.0
HAR	88.6 ± 0.1	67.6 ± 0.7	$\times$ 2.0	$\times$ 5.2
CDR	87.0 ± 0.2	75.8 ± 1.5	$\times$ 1.3	$\times$ 6.5
ALASCA (Different architecture and position of ACs)				
MLP	88.5 ± 0.3	76.3 ± 0.3	$\times$ 1.1	$\times$ **1.0**
MLP*	89.3 ± 0.1	80.8 ± 0.3	$\times$ 1.3	$\times$ 1.2
Bottleneck	89.2 ± 0.1	77.9 ± 0.4	$\times$ **1.0**	$\times$ **1.0**
Bottleneck*	$\mathbf{90.1 \pm 0.1}$	$\mathbf{81.4 \pm 0.3}$	$\times$ 1.1	$\times$ 1.2
Residual	89.0 ± 0.2	77.1 ± 0.3	$\times$ 1.1	$\times$ 1.1
Residual*	89.8 ± 0.2	77.8 ± 0.5	$\times$ 1.3	$\times$ 1.3

Table 3: Comparison of accuracy (%), GPU memory, and computation time. For HAR and CDR, we apply their official code. Without and with * denote attaching ACs at the end of only the third and all residual blocks of the backbone.

are computationally expensive to find noisy data or parameters. Table 3 shows that ALASCA consistently outperforms competing regularization-based methods for various noise types and provides efficient training in terms of computational memory and training time.

Effect of Auxiliary Classifiers. We further compare the performance of ALASCA with different architectures (2 layers MLP, residual block; He et al. 2016, bottleneck; Howard et al. 2017) and positions of ACs. Table 3 shows how the performance of ALASCA changes according to the architecture and position of ACs. We observe that using several ACs effectively regularizes the intermediate layer, resulting in improved generalization. This result supports our motivations for using ACs to regularize intermediate layers and thereby enhance the robustness against label noise. Furthermore, the architecture of ACs does not affect performance much. Among the three architectures, bottleneck classifiers achieve competitive performance with the smallest additional computation costs. From these results, we apply the bottleneck block as the AC for all experiments.

Component Analysis. Since our ALASCA is composed of three parts: (1) ALS; (2) ACs; and (3) EMA confidence, we perform a component analysis to understand which component is important for training robust feature extractors. Our experiments are conducted on CIFAR-10 under various noise distributions. Table 4 summarizes that each component is indeed effective, as the performance improves with each addition of a component. However, the most important factor for high performance is the combination of ALS and ACs, which enable effective LR on intermediate layers.

	Symm. 50	Asym. 40	Inst. 40
ULS	69.5 ± 1.1	78.7 ± 0.5	59.2 ± 0.6
+ AC	84.7 ± 0.2	87.2 ± 0.2	77.0 ± 1.0
ALS	82.0 ± 0.7	85.1 ± 0.2	74.8 ± 1.3
+ AC	86.6 ± 0.2	90.0 ± 0.3	78.8 ± 0.6
+ EMA	82.9 ± 0.8	86.9 ± 1.1	75.1 ± 2.1
ALASCA	88.0 ± 0.3	90.3 ± 0.3	81.4 ± 0.4
ALASCA*	$\mathbf{89.1 \pm 0.4}$	$\mathbf{90.6 \pm 0.2}$	$\mathbf{82.5 \pm 0.9}$

Table 4: Component analysis on each component of our proposed methods. ALASCA* denotes that result from ensemble of all classifiers during inference phase and the bold numbers indicate the best result.

Moreover, we verify that the performance of ALASCA improves using an ensemble of predictions from the main and ACs as we mentioned in Section 3.2.

4.4 Results on Real-world Datasets

Clothing1M (Xiao et al. 2015) contains one million clothing images obtained from online shopping websites with 14 classes and estimated noise level of 38.5% (Song et al. 2019). We apply ResNet50, which is widely used in previous studies (Liu et al. 2020; Kim et al. 2021b) on the Clothing1M dataset. Table 2 compares ALASCA to the SOTA methods on the Clothing1M dataset. ALASCA achieves a competitive performance with the SOTA baseline methods although using only a single network with lower computational costs. Furthermore, we observe that ELR+ with ALASCA (A-ELR+) realizes a new SOTA performance and verify that our proposed method also works well on real-world datasets. Additionally, we apply ALASCA on the (mini) WebVision dataset, a famous real-world dataset with label noise, and obtain similar results to those on Clothing1M. We report the detailed results in Appendix D.4.

5 Conclusion

In this paper, we provide a theoretical analysis that LS encourages LR, and build upon the resulting insights to propose an effective and practical framework, ALASCA. Based on the resulting theoretical motivation, we combine ALS, AC, and EMA confidence to efficiently enable adaptive LR. We experimentally show that ALASCA enhances the robustness of feature extractors and improves the performance of existing LNL methods on benchmark-simulated and real-world datasets. In future work, we believe that our approach will arise interest in designing a novel regularization strategy for feature extractors.

Acknowledgements

This work was supported by Center for Applied Research in Artificial Intelligence (CARAI) grant funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD) (UD190031RD).

References

Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M. S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. 2017. A closer look at memorization in deep networks. In *International Conference on Machine Learning*, 233–242. PMLR.

Cao, K.; Chen, Y.; Lu, J.; Arechiga, N.; Gaidon, A.; and Ma, T. 2021. Heteroskedastic and Imbalanced Deep Learning with Adaptive Regularization. In *International Conference on Learning Representations*.

Cheng, H.; Zhu, Z.; Li, X.; Gong, Y.; Sun, X.; and Liu, Y. 2020. Learning with instance-dependent label noise: A sample sieve approach. *arXiv preprint arXiv:2010.02347*.

Chorowski, J.; and Jaitly, N. 2017. Towards Better Decoding and Language Model Integration in Sequence to Sequence Models. In *INTERSPEECH*, 523–527.

Filiposka, M. Z.; Djuric, A. M.; and ElMaraghy, W. 2014. Complexity analysis for calculating the Jacobian matrix of 6DOF reconfigurable machines. *Procedia CIRP*, 17: 218–223.

Finlay, C.; Calder, J.; Abbasi, B.; and Oberman, A. 2018. Lipschitz regularized deep neural networks generalize and are adversarially robust. *arXiv preprint arXiv:1808.09540*.

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

Ghoshal, A.; Chen, X.; Gupta, S.; Zettlemoyer, L.; and Mehdad, Y. 2021. Learning Better Structured Representations Using Low-rank Adaptive Label Smoothing. In *International Conference on Learning Representations*.

Gouk, H.; Frank, E.; Pfahringer, B.; and Cree, M. J. 2021. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2): 393–416.

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; and Sugiyama, M. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Kim, T.; Ko, J.; Cho, S.; Choi, J.; and Yun, S.-Y. 2021a. FINE Samples for Learning with Noisy Labels. arXiv:2102.11628.

Kim, T.; Oh, J.; Kim, N.; Cho, S.; and Yun, S.-Y. 2021b. Comparing Kullback-Leibler Divergence and Mean Squared Error Loss in Knowledge Distillation. *arXiv preprint arXiv:2105.08919*.

Lee, H.; Shin, S.; and Kim, H. 2021. ABC: Auxiliary Balanced Classifier for Class-imbalanced Semi-supervised Learning. *Advances in Neural Information Processing Systems*, 34.

Lee, K.-H.; He, X.; Zhang, L.; and Yang, L. 2018. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5447–5456.

Li, J.; Socher, R.; and Hoi, S. C. 2020. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*.

Li, W.; Dasarathy, G.; and Berisha, V. 2020. Regularization via Structural Label Smoothing. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, 1453–1463. PMLR.

Liu, S.; Niles-Weed, J.; Razavian, N.; and Fernandez-Granda, C. 2020. Early-learning regularization prevents memorization of noisy labels. *arXiv preprint arXiv:2007.00151*.

Lukasik, M.; Bhojanapalli, S.; Menon, A.; and Kumar, S. 2020. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, 6448–6458. PMLR.

Mirzasoleiman, B.; Cao, K.; and Leskovec, J. 2020. Coresets for robust training of neural networks against noisy labels. *arXiv preprint arXiv:2011.07451*.

Müller, R.; Kornblith, S.; and Hinton, G. 2019. When does label smoothing help? *arXiv preprint arXiv:1906.02629*.

Nesser, H.; Jacob, D. J.; Maasakkers, J. D.; Scarpelli, T. R.; Sulprizio, M. P.; Zhang, Y.; and Rycroft, C. H. 2021. Reduced-cost construction of Jacobian matrices for high-resolution inversions of satellite observations of atmospheric composition. *Atmospheric Measurement Techniques*, 14(8): 5521–5534.

Nettleton, D. F.; Orriols-Puig, A.; and Fornells, A. 2010. A study of the effect of different types of noise on the precision of supervised learning techniques. *Artificial intelligence review*, 33(4): 275–306.

Nguyen, D. T.; Mummadi, C. K.; Ngo, T. P. N.; Nguyen, T. H. P.; Beggel, L.; and Brox, T. 2020. SELF: Learning to Filter Noisy Labels with Self-Ensembling. In *International Conference on Learning Representations*.

Pereyra, G.; Tucker, G.; Chorowski, J.; Kaiser, ; and Hinton, G. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions. *arXiv preprint arXiv:1701.06548*.

Sokolić, J.; Giryes, R.; Sapiro, G.; and Rodrigues, M. R. 2017. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16): 4265–4280.

Song, H.; Kim, M.; Park, D.; and Lee, J.-G. 2019. How does Early Stopping Help Generalization against Label Noise? *arXiv preprint arXiv:1911.08059*.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Tibshirani, R. J. 2014. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1): 285–323.

Wang, X.; Du, P.; and Shen, J. 2013. Smoothing splines with varying smoothing parameter. *Biometrika*, 100(4): 955–970.

Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; and Bailey, J. 2019. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 322–330.

Wei, C.; and Ma, T. 2019a. Data-dependent sample complexity of deep neural networks via lipschitz augmentation. *arXiv preprint arXiv:1905.03684*.

Wei, C.; and Ma, T. 2019b. Improved sample complexities for deep networks and robust classification via an all-layer margin. *arXiv preprint arXiv:1910.04284*.

Wei, J.; Liu, H.; Liu, T.; Niu, G.; and Liu, Y. 2021a. Understanding Generalized Label Smoothing when Learning with Noisy Labels. *arXiv preprint arXiv:2106.04149*.

Wei, J.; Liu, H.; Liu, T.; Niu, G.; Sugiyama, M.; and Liu, Y. 2021b. To Smooth or Not? When Label Smoothing Meets Noisy Labels. *Learning*, 1(1): e1.

Xia, X.; Liu, T.; Han, B.; Gong, C.; Wang, N.; Ge, Z.; and Chang, Y. 2021. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*.

Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; and Wang, X. 2015. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2691–2699.

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

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization. arXiv:1611.03530.

Zhang, H.; and Yao, Q. 2020. Decoupling Representation and Classifier for Noisy Label Learning. *arXiv preprint arXiv:2011.08145*.

Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; and Ma, K. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3713–3722.

Zhang, Z.; and Sabuncu, M. R. 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. In *32nd Conference on Neural Information Processing Systems (NeurIPS)*.

Zheltonozhskii, E.; Baskin, C.; Mendelson, A.; Bronstein, A. M.; and Litany, O. 2022. Contrast to divide: Self-supervised pre-training for learning with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1657–1667.