Student Loss: Towards the Probability Assumption in Inaccurate Supervision

Shuo Zhang, Jian-Qing Li, Hamido Fujita, *Life Senior Member, IEEE,* Yu-Wen Li, Deng-Bao Wang,
Ting-Ting Zhu, Min-Ling Zhang, *Senior Member, IEEE,* and Cheng-Yu Liu, *Senior Member, IEEE*

Abstract—Noisy labels are often encountered in datasets, but learning with them is challenging. Although natural discrepancies between clean and mislabeled samples in a noisy category exist, most techniques in this field still gather them indiscriminately, which leads to their performances being partially robust. In this paper, we reveal both empirically and theoretically that the learning robustness can be improved by assuming deep features with the same labels follow a student distribution, resulting in a more intuitive method called student loss. By embedding the student distribution and exploiting the sharpness of its curve, our method is naturally data-selective. This ability makes clean samples aggregate tightly in the center, while mislabeled samples scatter, even if they share the same label. Additionally, we employ the metric learning strategy and develop a large-margin student (LT) loss for better capability. It should be noted that our approach is the first work that adopts the prior probability assumption in feature representation to decrease the contributions of mislabeled samples. This strategy can enhance various losses to join the student loss family, even if they have been robust losses. Experiments demonstrate that our approach is more effective in inaccurate supervision. Enhanced LT losses significantly outperform various state-of-the-art methods in most cases. Even huge improvements of over 50% can be obtained under certain conditions. An implementation of the main codes is available at https://github.com/Zhangshuojackpot/Student-Loss.

Index Terms—Learning with Noisy label, Robust Loss Function, Deep Learning

1 Introduction

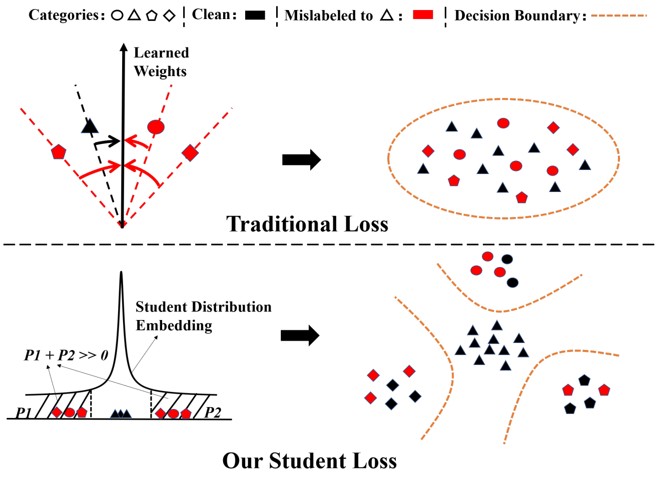


Fig. 1. The comparison between employing the traditional loss and our student loss in inaccurate supervision. The traditional loss attempts to gather clean and mislabeled samples and obtains messy categorical clusters. However, since we introduce a long-tail student distribution to feature representation, our student loss obtains distinguishable categorical clusters even if mislabeled samples exist.

Recent developments in supervised deep neural networks (DNNs) have considerably increased the performance of state-of-the-art (SOTA) models in various applications. These successes are highly dependent on the emergence of large-scale datasets that have been carefully labeled. Nevertheless, labeling precise annotations for training is time-consuming and prone to mistakes (even high-quality datasets, such as ImageNet [1], include erroneous labels [2]). Therefore, inaccurate supervision, particularly in learning with noisy labels, is a critical issue in practical deep learning tasks [3]. Numerous approaches have been suggested sequentially to address this issue, including: 1) Robust Architecture [4], [5], [6], [7], [8], [9], [10], [11], in which some novel structures of DNNs are designed to limit the mislabeled samples; 2) Robust Regularization [12], [13], [14], [15], [16], [17], in which some additional constraints should be met during

- *Shuo Zhang is with the School of Instrument Science and Engineering, the State Key Laboratory of Bioelectronics, the School of Biological Science and Medical Engineering, Southeast University. Nanjing 210096, China. E-mail: zs_techo@seu.edu.cn.*
- *Jian-Qing Li, Yu-Wen Li and Cheng-Yu Liu are with the State Key Laboratory of Bioelectronics, School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China. E-mail: ljq@seu.edu.cn., liyuwen@seu.edu.cn., chengyu@seu.edu.cn.*
- *Hamido Fujita is with HUTECH University, Ho Chi Minh City, Vietnam, and with the Andalusian Research Institute in Data Science and Computational Intelligence (DaSCI), University of Granada, Granada, Spain, and also with Regional Research Center, Iwate Prefectural University, Iwate, Japan. E-mail: h.fujita@hutech.edu.vn., HFujita-799@acm.org.*
- *Deng-Bao Wang and Min-Ling Zhang are with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and also with the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China. E-mail: wangdb@seu.edu.cn., zhangml@seu.edu.cn.*
- *Ting-Ting Zhu is with the Department of Engineering Science, University of Oxford, Oxford OX3 7DQ, U.K.. E-mail: tingting.zhu@eng.ox.ac.uk.*

This work was supported by the National Key Research and Development Program of China under Grant 2019YFE0113800, the National Natural Science Foundation of China under Grants 62171123, 62211530112, 62071241, 62001111 and 81871444, the Natural Science Foundation of Jiangsu Province of China under Grants BK20192004 and BK20200364.
(Corresponding author: Jian-Qing Li and Cheng-Yu Liu.)

convergence; 3) Sample Selection [18], [19], [20], [21], [22], [23], [24], [25], [26], in which clean samples are picked up as much as possible for training. 4) Robust Loss Design [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], in which some new robust loss functions are proposed to learn with noisy labels. In comparison to alternative techniques that may suffer from imprecise noise estimates or complicated training procedures, applying the robust loss is simpler and more effective, so that is also the main focus of this paper.

Generally speaking, a discriminative loss is encouraged by congregating the samples with the same labels, which can be achieved by evaluating the sample similarity on a specific metric (such as cosine distance, Mahalanobis distance, *etc.*). The angle between the feature vector and the learned categorical weight vector in the last layer is usually chosen as this metric. However, this kind of thinking is largely valid but falls short when the label is inaccurate. As shown in Fig. 1, if we apply a traditional loss to learn a noisy category for classification, not only does the angle between the clean sample and the learned weight vector decrease, but that between the mislabeled sample and the learned weight vector also decreases. It generates discrepancies and finally leads to an inseparable categorical cluster. To overcome the defects of losses in inaccurate supervision, researchers have made many attempts: [27] compared Categorical Cross-Entropy (CCE) with Mean Absolute Error (MAE) loss and concluded that MAE is more noise-resistant due to its data-equal characteristic. This result prompted [28] to propose Generalized Cross-Entropy (GCE), which can be seen as a combination of CCE and MAE. Furthermore, [29] offered the Negative Learning for Noisy Label (NLNL) strategy, which introduced a three-stage pipeline for filtering the noisy data. [30] proposed Symmetric Cross-Entropy (SCE), which was a robust variant of CCE that combined CCE with Reverse Cross-Entropy (RCE). Extensively, [31] categorized current losses as "Active" or "Passive" and proposed Active Passive Loss (APL). This technique combines active losses that induce overfitting (such as CCE) with a passive loss that causes underfitting (such as MAE) to achieve optimal performance. Meanwhile, [32] employed the curriculum loss (CL), which is a surrogate loss of the 0–1 loss function and provided a tight upper bound that can easily be extended to multi-class classification. Recently, [33] reported the Joint Negative and Positive Learning (JNPL) and claimed it can be regarded as an improved approach of NLNL. [34] practiced the Jensen-Shannon Divergence (JS) loss and its generalized version, which trained the samples by Jensen-Shannon Divergence.

Despite numerous efforts, these SOTA methods do not get rid of the thinking stereotype that still aggregates samples with the same labels as much as possible, although some mislabeled samples exist. It also leads to them being partially robust. In fact, natural discrepancies between clean and mislabeled samples of the same label hinder the traditional attempt but trigger us to achieve unsupervised distinctions with prior assumptions. Specifically, by considering deep features under the same label to follow a long-tail student distribution in the penultimate hidden space, in this paper, we propose a more intuitive and effective method called student loss. As shown in Fig. 1, since the curve of the student distribution is extremely steep in a particular region, the edge probabilities ($P1$ and $P2$ in Fig. 1) are reserved. It allows our approach to have a naturally data-selective capability and can be applied to fight against the inconsistency produced by the labeling errors. Additionally, we introduce a hyperparameter to encourage wider inter-class distance and further propose a large-margin student (LT) loss. Following our approach, intra-class clean samples can aggregate tightly in the center, while mislabeled samples scatter at the edge, achieving an unsupervised clean/mislabeled sample partition. *It should be noted that the student loss is the first research which introduces an assumption of prior probability distribution in the hidden space to improve the performance in inaccurate supervision. Moreover, various losses, even SOTA robust losses, can be further strengthened by our method.* Our major contributions can be summarized as follows:

- We provide an insight into the probability distribution of deep features and not only **empirically** but also **theoretically** point out that the robustness of learning with noisy labels can be improved by assuming the samples with the same label to follow the student distribution.
- Based on this perspective, we propose the **student loss**. It is data-selective by embedding the student distribution, causing clean intra-class samples to concentrate neatly while mislabeled samples disperse, even if their labels are uniform. Furthermore, we employ some strategies from metric learning and develop its **large-margin** version.
- **Various losses** can be enhanced by our approach. Experiments on both benchmark and real-world datasets demonstrate that LT losses can achieve better performances than SOTA approaches in inaccurate supervision.

2 Related Work

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

1) Robust Architecture. These approaches aim to employ a noise adaptation layer on top of a DNN to learn the label transition process or create a dedicated architecture to support more varied types of label noise. As such, Webly learning [4] first taught the underlying DNN to retrieve only simple instances. The confusion matrices of all training instances were utilized as the initial weight W of the noise adaptation layer. The noise model [5] set W to an identity matrix and added a regularizer to diffuse W during DNN training. In [6], Dropout regularization was applied to the adaptation layer, whose output was normalized by softmax to implicitly diffuse W. Similarly, the s-model [7] was proposed for the dropout noise model but without dropout. The c-model [7] was regarded as an extension of the s-model, which was more realistic than the symmetric and asymmetric noises. The Noisy labels Neural-Network (NLNN) algorithm [8] applied the Expectation-Maximization (EM) method to execute the E-step to estimate the noise transition matrix and the M-step to backpropagate the DNN. Additionally, several research projects built specific noise-handling structures. Masking [9] was a human-aided method of communicating the human understanding of erroneous label transitions. The faulty transition explored by humans was used to confine noise modeling. On the other hand, to anticipate the noise type and label transition probability, probabilistic noise modeling [10] controlled two separate networks. The contrastive-additive noise network [11] was recently presented to compensate for inaccurately estimated label transition probabilities. This network introduced the novel notion of quality embedding to characterize the reliability of noisy labels.

2) Robust Regularization. These approaches aim to prevent a DNN from overfitting false-labeled instances by creating some training restrictions. The primary benefit of this group is that it can easily adapt to new contexts by incorporating very few changes.

As such, Bilevel learning [12] offered a different tactic by presenting a bilevel optimization strategy to regularize the overfitting of a model using a clean validation dataset. This approach varied from the standard one in that the regularization constraint was itself an optimization issue. Mini-batch-level weight adjustments and validation-set error minimization were used to rein in overfitting. Equally, in [13], it was assumed that several annotators existed, and a regularized EM-based technique was introduced to model the label transition probability. Besides, fine-tuning a pretrained model yielded a large gain in resilience compared with models trained from scratch [14]. This was because the universal representations learned during pretraining prevented the model parameters from being updated in the incorrect direction by noisy labels. For adapting to clean and noisy labels, respectively, robust early learning [15] categorized factors as either crucial or noncritical, and only noncritical updates were penalized. To increase resistance to label noise, PHuber [16] suggested a composite loss-based gradient clipping as an alternative to traditional gradient clipping. By randomly labeling the open-set instances, ODLN [17] employed the open-set auxiliary data and eliminated the overfitting of noisy labels.

3) Sample Selection. These approaches aim to isolate the most likely clean samples for optimization. As such, [18] offered MentorNet, which was considered a curriculum-based approach for learning the most likely correct samples. Decouple [19] recommended uncoupling update frequency from update methodology. Hence, two DNNs were kept in parallel and only modified when the examples were judged to have a disagreement. Similarly, in Co-teaching [20] and Co-teaching+ [21], two DNNs were kept. One DNN chose a predetermined number of low-loss samples and input them to the other DNN for training. Co-teaching+ added decoupling disagreement to co-teaching. INCV [22] divided noisy training data at random and then applied cross validation to classify clean examples while getting rid of mislabeled examples in training. JoCoR [23] practiced co-regularization to lower the diversity between two DNNs, bringing together their predictions. DivideMix [24] employed two-component and one-dimensional Gaussian mixture models to fit the loss values of samples and turned noisy samples into labeled and unlabeled sets. Then, a semi-supervised technique called MixMatch [25] was introduced for classification. RoCL [26] similarly followed a two-stage learning strategy: first, supervised training on selected clean examples, and second, semi-supervised learning on relabeled noisy samples under self-supervision. It computed the exponential moving average of training loss for selection and relabeling. Although learning through sample selection is effective in most cases, it produces a large amount of accumulated error when there are numerous ambiguous labels in the training data.

4) Robust Loss Design. These approaches aim to adjust the loss value according to the confidence of a given loss (or label) by some strategies or design a new loss function for inaccurate supervision. Robust losses typically include a constraint to penalize predictions with a low degree of confidence that are most likely caused by noisy samples. This subject is the most pertinent to our work. Since some robust losses have been discussed in Sec. 1, here we only report some loss adjustment techniques. As such, it can be categorized into three groups: 1) loss correction. This approach multiplied the predicted label transition probability to adjust the loss. Backward [35] first made an approximation of the noise transition matrix by employing the softmax output of the DNN trained without loss correction. Subsequently, it refreshed the DNN with a revised loss based on the estimated matrix. Forward [35] combined the softmax output of a DNN before applying the loss function. Recently, T-revision [36] offered a technique that inferred the transition matrix without anchor points. To circumvent directly estimating the noisy class posterior, dual T [37] factorized the matrix into the product of two easy-to-estimate matrices. 2) loss reweighting. This approach gave smaller weights to the mislabeled examples and greater weights to the clean examples. Active bias [38] prioritized uncertain examples with inconsistent label predictions by applying their prediction variances as training weights. DualGraph [39] practiced graph neural networks, reweighted the examples by the structural relations among labels, and eliminated the abnormal noise examples. 3) label refurbishment. This approach repaired a noisy label to avoid overfitting incorrect labels. Bootstrapping [40] was the first method that provided the concept of label refurbishment to update the label of training examples. It built a more coherent network that improved its ability to evaluate the consistency of noisy labels by the label confidence via cross validation. D2L [41] developed a DNN with a dimensionality-driven learning strategy to prevent overfitting to the false label. SELFIE [42] proposed a paradigm of refurbishable examples that can be revised with high precision. The main notion was to regard examples with consistent label predictions as refurbishable because of the learner's perceptual constancy.

3 Student Loss

3.1 Preliminaries

Assuming a K-class classification task, a posterior probability of true class $z \in [1, K]$ can be obtained by the softmax function as

$$p(z|x_i) = \frac{e^{f_z(x_i)}}{\sum_{k=1}^{K} e^{f_k(x_i)}}, \tag{1}$$

where x_i represents the deep feature vector of the i-th sample in the training set. If x_i is the input vector of the last fully connected layer, an affinity score $f_k(x_i)$ generates to a linear transform written as

$$f_k(x_i) = w_k^T x_i + b_k, \tag{2}$$

where w_k and b_k represent weights and bias corresponding to k-th class. Then if we employ CCE for training, the loss L can be written as

$$L = -\sum_{k=1}^{K} t_k \log(\frac{e^{f_k(x_i)}}{\sum_{j=1}^{K} e^{f_j(x_i)}}), \tag{3}$$

where t_k represents the k-th value of the one-hot label.

3.2 Robust Student Loss

Generally, let $p(x_i)$ represent the probability that sample x_i occurs, and it can be written as

$$p(x_i) = \sum_{k=1}^{K} p(x_i|k)p(k) = \Delta x \sum_{k=1}^{K} f_X^k\left(x_i|\theta\right) p(k), \tag{4}$$

where $f_X^k\left(x_i|\theta\right)$ is the value of the specific probability density function of k-th class at point x_i. θ is the parameter of the distribution function. $p(k)$ represents the prior probability of k-th class. After that, the posterior probability $p(z|x_i)$ can be written as

$$p(z|x_i) = \frac{f_X^z\left(x_i|\theta\right) p(z)}{\sum_{k=1}^{K} f_X^k\left(x_i|\theta\right) p(k)}. \tag{5}$$

In [43], the authors proposed the GM loss by assuming f to a Gaussian distribution function and discovered that it performs well on the clean dataset. However, the Gaussian function makes nearly all of the samples compulsorily distribute within three times the variance surrounding the mean, which is defective in inaccurate supervision. In order to obtain a more tolerant representation, we employ a long-tail student distribution to exhibit the deep features. Under this line, the f can generate into

$$\begin{aligned} f_X^z(x_i|\theta) &= \mathcal{T}(x_i|n_z,\mu_z,\Sigma_z) \\ &= \frac{\Gamma(\frac{n_z+l}{2})}{\Gamma(\frac{n_z}{2})n_z^{\frac{l}{2}}\pi^{\frac{l}{2}}}|\Sigma|^{-\frac{1}{2}}[1+\frac{D(x_i)}{n_z}]^{-(\frac{n_z+l}{2})}, \end{aligned} \tag{6}$$

$$D(x_i) = (x_i-\mu_z)^T\Sigma^{-1}(x_i-\mu_z), \tag{7}$$

where n, μ and Σ represent the freedom degree, the mean, and the covariance matrix of each class, respectively. They are three trainable parameters introduced because of the student distribution assumption. l represents the number of feature dimensions, and D represents the Mahalanobis distance between deep features.

For the calculability and simplicity of optimization, we rewrite Eq. (6) and Eq. (7) as follows: 1) We limit x and μ to make them lie on the ℓ_2-norm ball. 2) We introduce a hyperparameter φ to confine the lower bound of the freedom degree and ensure that the probability at curve edge is not excessively large. In the paper, it is set to 0.1. 3) We eliminate the constant $\pi^{\frac{l}{2}}$ and replace l with $\ln l$ to ensure the computed correction of extreme high-dimension inputs. 4) We assume Σ is the identity matrix Λ and the prior probability $p(k) = 1/K$. As such, the revised class probability function $\widehat{f}$ and the posterior probability $p_t(z|x_i)$ can be written as:

$$\begin{aligned} \widehat{f}_X^z(x_i|\theta) &= \mathcal{T}'(x_i|n_z,\mu_z,\Lambda_z) \\ &= \frac{\Gamma(\frac{\widehat{n}+\ln l}{2})}{\Gamma(\frac{\widehat{n}}{2})\widehat{n}^{\frac{\ln l}{2}}}[1+\frac{\widehat{D}(x_i)}{\widehat{n}}]^{-(\frac{\widehat{n}+\ln l}{2})}, \end{aligned} \tag{8}$$

$$\widehat{n} = |n_z| + \varphi, \tag{9}$$

$$\widehat{D}(x_i) = (\frac{x_i}{\|x_i\|_2}-\frac{\mu_z}{\|\mu_z\|_2})^T(\frac{x_i}{\|x_i\|_2}-\frac{\mu_z}{\|\mu_z\|_2}), \tag{10}$$

$$p_t(z|x_i) = \frac{\widehat{f}_X^z(x_i|\theta)}{\sum_{k=1}^K \widehat{f}_X^k(x_i|\theta)}. \tag{11}$$

According to the above distribution embedding strategy, we propose our student loss L_t. Overall, it includes two parts:

$$L_t = L_{tds} + \lambda L_C. \tag{12}$$

The first term L_{tds} represents a discriminative loss. Various losses can be further employed as L_{tds}. The second term L_C represents the center loss [44], which can be written as

$$L_C = ||x_i - \mu_z||_2^2. \tag{13}$$

Attributed to L_C, the student loss can detect the mislabeled samples. λ represents a weighting hyperparameter.

In other words, our strategy is universal. Many popular losses, even the SOTA robust losses, can be strengthened to join the student loss family. For example, a student loss $L_{t_{cce}}$ based on CCE can be written as

$$L_{t_{cce}} = -\sum_{k=1}^K t_k\log(p_t(k|x_i)) + \lambda L_C. \tag{14}$$

A student loss $L_{t_{sce}}$ based on SCE can be written as

$$\begin{aligned} L_{t_{sce}} =& -\alpha\cdot\sum_{k=1}^K t_k\log(p_t(k|x_i)) - \beta\cdot\sum_{k=1}^K p_t(k|x_i)\log(t_k) \\ &+\lambda L_C, \end{aligned} \tag{15}$$

where α and β are the weighting hyperparameters.

Furthermore, from a metric learning standpoint, the generalization of a deep model can be enhanced by increasing the inter-class margin [45], [46]. Similar to [43], an extra hyperparameter ϵ is introduced to restrain $\widehat{D}$. Accordingly, $\widehat{f}$ can generate into

$$\widehat{g}_X^z(x_i|\theta) = \frac{\Gamma(\frac{\widehat{n}+\ln l}{2})}{\Gamma(\frac{\widehat{n}}{2})\widehat{n}^{\frac{\ln l}{2}}}[1+\frac{\widehat{D}(x_i)\cdot(1+\epsilon)}{\widehat{n}}]^{-(\frac{\widehat{n}+\ln l}{2})}. \tag{16}$$

We replace $\widehat{f}$ in Sec. 3.2 with $\widehat{g}$, so that further develop their large-margin versions.

3.3 Theoretical Illustration

Proposition 1. *When the freedom degree n converges to a positive infinity, the unsimplified student loss degenerates into the GM loss.*

It is commonly accepted that the student distribution $\mathcal{T}(x_i|n_{z_i},\mu_{z_i},\Sigma_{z_i})$ becomes the Gaussian distribution $\mathcal{N}(x_i|\mu_{z_i},\Sigma_{z_i})$ when $n \to +\infty$. In other words, the GM loss is a specific case of the unsimplified student loss that we propose. Strikingly, the introduced n is employed to change the sharpness of the curve adaptively, making it indispensable in inaccurate supervision.

Proposition 2. *When the sample number converges to a positive infinity, the accuracy (Acc) of k-th class can be calculated as*

$$Acc = \frac{\underbrace{\int\cdots\int}_{D}\mathcal{T}'(x|n_k,\mu_k,\Lambda_k)\mathrm{d}A}{\underbrace{\int\cdots\int}_{D}\mathcal{T}'(x|n_k,\mu_k,\Lambda_k)\mathrm{d}x}, \tag{17}$$

where A represents the area enclosed by the decision boundary of k-th class. D represents the dimension of deep features.

Note that, we assume $\widehat{f}_X^k(x|\theta)$ as $\mathcal{T}'(x|n_k,\mu_k,\Lambda_k)$ and deduce the posterior probability $p(k|x)$ by the Bayesian formula. As a result, Acc satisfies the rule of the minimal Bayesian error rate. It reveals that when the number of training samples is large enough, an theoretical accuracy of deep learning in which a statistical model embeds can be derived.

Proposition 3. *When applying the student loss, the gradients provided by the mislabeled samples in the convergence are relatively limited.*

This proposition explains the effect of our approach from a gradient perspective. According to [47], assuming x_m represents a mislabeled sample and employ CCE to train a network F, its gradient $\nabla L_{cce}(\Theta_F)$ can be written as

$$\nabla L_{cce}(\Theta_F) = \underbrace{(p_{x_m} - y_{x_m})}_{scale\ term}\cdot\nabla F_{x_m}(\Theta_F), \tag{18}$$

where p_{x_m} and y_{x_m} denote the prediction and the label of x_m, respectively. Θ denotes the trainable parameters in F. As can be seen, $p_{x_m} - y_{x_m}$ is a significant term in the inaccurate supervision. This part represents a small value when the label is correct but a large one when it is incorrect. Accordingly, the mislabeled sample provides a much larger gradient than the clean sample accompanied by the convergence process, which leads to poor performance.

In contrast, while employing the student loss (taking L_{tcce} as an example), $F(x_m)$ can further generate as $F'(\mathcal{T}(x_m))$. F' denotes the projection before the student distribution embedding. As such, its gradient $\nabla L_{tcce}(\Theta_{F'})$ can be written as:

$$\begin{aligned}\nabla L_{tcce}(\Theta_{F'}) &= \frac{\Gamma(\frac{\widehat{n}+\ln l}{2})}{\Gamma(\frac{\widehat{n}}{2}) \cdot \widehat{n}^{\frac{\ln l}{2}}} \cdot (-\frac{\widehat{n}+\ln l}{2}) \cdot (p_{x_m} - y_{x_m}) \\ &\quad \cdot (1 + \frac{\widehat{D}_{x_m}}{\widehat{n}})^{-(\frac{\widehat{n}+\ln l}{2}+1)} \cdot \frac{1}{\widehat{n}} \cdot \nabla \widehat{D}(\Theta_{F'}) \\ &= \underbrace{-H(\widehat{n}) \cdot \frac{p_{x_m} - y_{x_m}}{(\widehat{D}_{x_m} + \widehat{n})^{\frac{\widehat{n}+\ln l}{2}+1}}}_{adaptive\ scale\ term} \cdot \nabla \widehat{D}(\Theta_{F'}),\end{aligned} \tag{19}$$

$$H(\widehat{n}) = \widehat{n}^{\frac{\widehat{n}+\ln l}{2}+1} \cdot \frac{\Gamma(\widehat{n}+\ln l)(\widehat{n}+\ln l)}{2 \cdot \Gamma(\frac{\widehat{n}}{2}) \cdot \widehat{n}^{\frac{\ln l}{2}+1}}, \tag{20}$$

We observe that $p_{x_m} - y_{x_m}$ is limited by $\widehat{D}_{x_m}$ in the student distribution. The mislabeled sample x_m also produce a larger $\widehat{D}_{x_m}$, making $\frac{p_{x_m} - y_{x_m}}{(\widehat{D}_{x_m}+\widehat{n})^{\frac{\widehat{n}+\ln l}{2}+1}}$ change to be a small value. Furthermore, the term of $H(\widehat{n})$ generates an adaptive scale. Shifting $\widehat{n}$ can dynamically adjust the gradient, providing a more tolerant convergence. Obviously, this proposition theoretically demonstrates the effectiveness of our approach from a gradient weighting perspective. It is also consistent with our motivation for introducing the long tail of the student distribution to hold the mislabeled samples.

3.4 Discussions

Why Student Distribution? It is commonly documented that loss functions encourage gathering naturally similar samples while dispersing dissimilar samples. Usually, traditional losses achieve it by learning a categorical template (weight vectors) and directly minimizing the cosine distance between the sample and the template. Since categorical information can only be transmitted by the label, mislabeled samples produce intra-class inconsistency and finally result in the messy cluster shown in Fig. 1. In contrast to previous approaches, we rethink inaccurate supervision from the perspective of probability distribution and define the deep features of one noisy category to follow the student distribution. By this assumption, the prior huge disparities in the distribution can tolerate this inconsistency. In other words, the long-tail property of the student distribution can "absorb" most mislabeled samples and make different categories recognizable, even if they share the same label. Therefore, our student loss can minimize intervals among the intra-class clean samples under inaccurate supervision tasks.

TABLE 1
Test accuracy (%) of the GM loss and our LT loss on benchmark datasets with various rates of symmetric noisy labels. The best results are in **bold**.

Methods	Datasets			
	MNIST		CIFAR10	
	$\eta = 0.2$	$\eta = 0.6$	$\eta = 0.2$	$\eta = 0.6$
CCE	90.24	49.44	77.93	41.11
GM Loss ($\Sigma = \Lambda$) [43]	94.02	73.26	77.92	47.18
GM Loss [43]	90.71	58.87	78.23	70.02
LT Loss	**98.92**	**97.91**	**90.20**	**80.06**

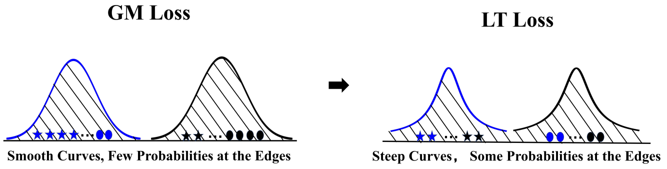
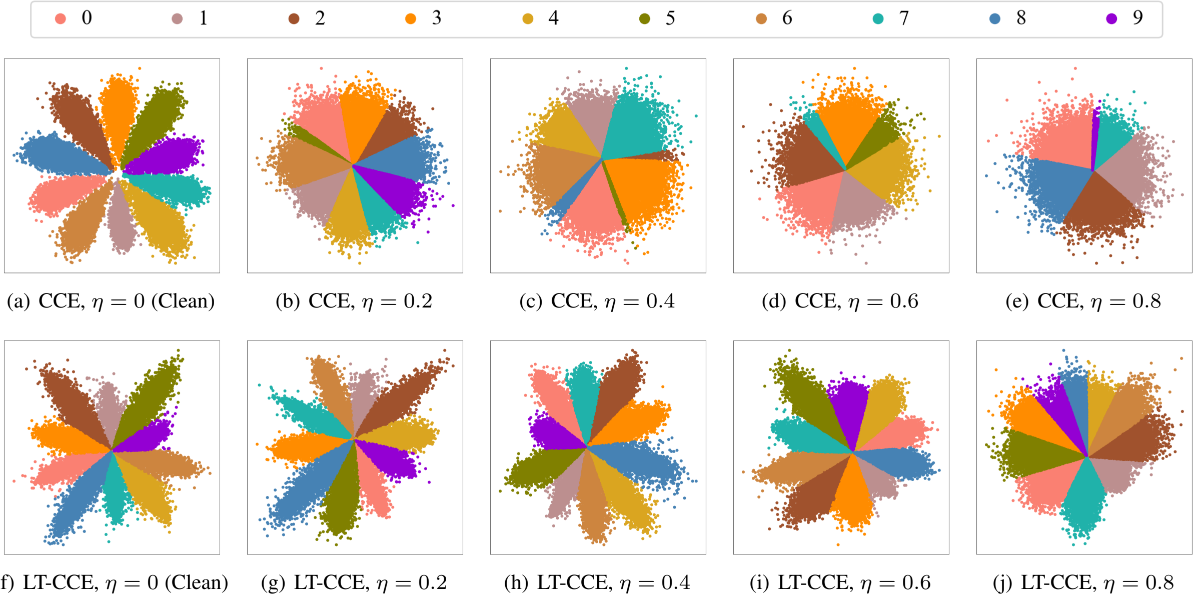

GM Loss
LT Loss
Smooth Curves, Few Probabilities at the Edges
Steep Curves, Some Probabilities at the Edges

Fig. 2. The differences between the GM loss and our LT loss in inaccurate supervision. Samples with the same colors have identical labels, and samples with the same shapes belong to identical categories. Our LT loss has steeper curves of categorical distributions and higher probabilities at their edges, which makes it more effective in this field.

Why Attach L_C? According to the analysis in Sec. 3.2, we can design L_{tds} with student distribution embedding to resist the label noise. As for L_C in the formulation, three impacts are considered: 1) Similar to [43], it acts as a likelihood regularization term to limit the distance between the outlier and the class centroid μ_z. 2) Since L_C is directly minimized during the training procedure, it can assist the classification and accelerate the convergence. 3) Last but most importantly, as the mislabeled sample is actually closer to the centroid of its natural category, L_C of the clean sample and that of the mislabeled sample are extremely distinctive, which can be utilized to identify and even revise incorrect labels. In our experiments, we empirically demonstrate that L_C plays a significant role and is indispensable (see Secs. 4.1 and 4.4).

Generalization of LT Loss. The traditional thinking to robust loss design is to generate a specific loss for addressing the problems in learning with noisy labels. Different from it, we make assumptions about the feature representation and directly employ the prior distribution to construct the loss function. This strategy results in many losses that were previously sensitive to label noise becoming robust. In other words, we not only propose a robust loss in this paper, but more importantly, we propose **a paradigm to make common losses robust**. We have demonstrated that our approach can enhance many losses, even robust losses, to suppress label noise (see Sec. 4.2). The generalization is regarded as one of evidences for the advancement of our method.

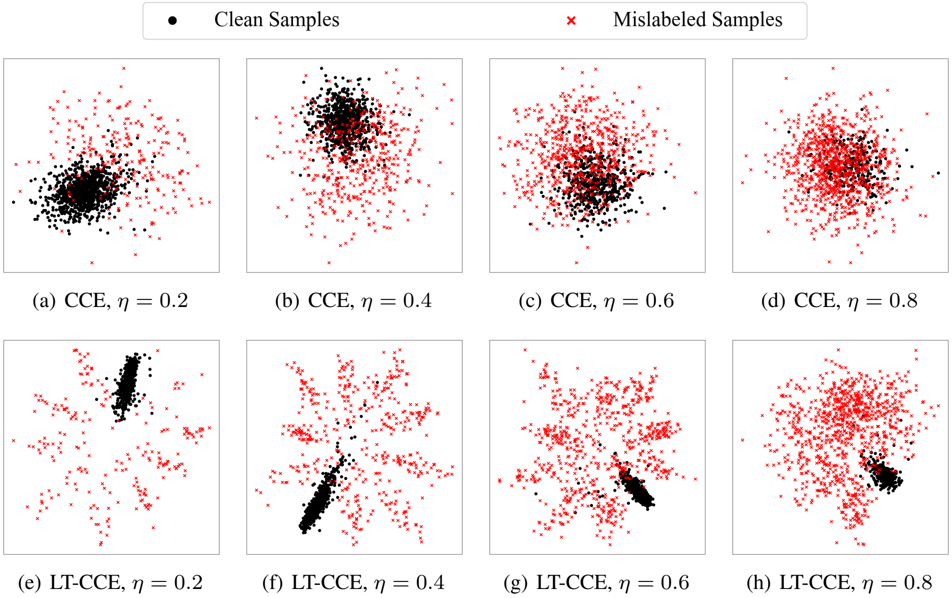
GM Loss versus LT Loss. The GM loss [43] and the LT loss both consider feature representation from a probability perspective, and the LT loss draws on the ideas of the GM loss in some ways. Nonetheless, they have the following distinctions: 1) According to Proposition 1 in Sec. 3.3, the GM loss is a **particular case** of unsimplified LT loss. Since Σ and n are both the parameters for adjusting the tightness of samples, the modification, preserving n and simplifying Σ to the identity matrix, has no effect on this generalization. Accordingly, our LT loss is more universal than the GM loss. 2) Since we regulate the norms of the features and the means in our approach, the LT loss still metrics the **cosine distance** for classification in the penultimate feature space, while the GM loss metrics the Mahalanobis distance. 3) Notably, as the term of D_{x_m} in the GM loss also hinders the increase of $p_{x_m} - y_{x_m}$ when training with mislabeled samples, this method can also improve performance to a certain extent. Nevertheless, its gradient is not flexible, making it not as efficient as our strategy. In other words, the **smooth** Gaussian function makes noisy samples



(a) CCE, $\eta = 0$ (Clean) (b) CCE, $\eta = 0.2$ (c) CCE, $\eta = 0.4$ (d) CCE, $\eta = 0.6$ (e) CCE, $\eta = 0.8$

(f) LT-CCE, $\eta = 0$ (Clean) (g) LT-CCE, $\eta = 0.2$ (h) LT-CCE, $\eta = 0.4$ (i) LT-CCE, $\eta = 0.6$ (j) LT-CCE, $\eta = 0.8$

Fig. 3. Feature representations using CCE and LT-CCE under the various noise rates η of symmetric noise. Our approach can harvest more robust clusters under $\eta > 0$



(a) CCE, $\eta = 0.2$ (b) CCE, $\eta = 0.4$ (c) CCE, $\eta = 0.6$ (d) CCE, $\eta = 0.8$

(e) LT-CCE, $\eta = 0.2$ (f) LT-CCE, $\eta = 0.4$ (g) LT-CCE, $\eta = 0.6$ (h) LT-CCE, $\eta = 0.8$

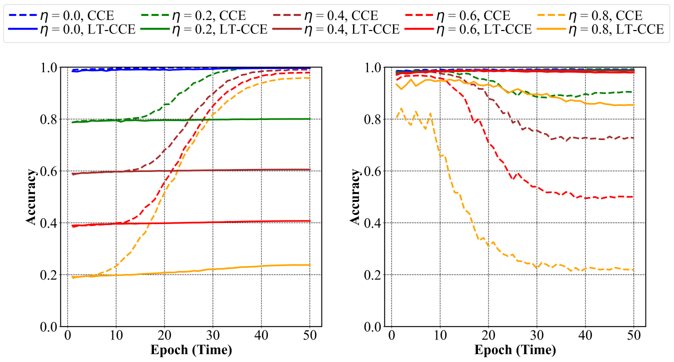
Fig. 4. Feature representations of noisy category '1' in MNIST using CCE and LT-CCE under the various noise rates η of symmetric noise. It is obvious that LT-CCE can distinguish between clean samples and mislabeled samples even if they share the same label.

dispersed within three times the variance surrounding the mean, leading to chaos within the cluster. On the contrary, the student function keeps some probabilities at the edge and maintains the sharpness of the curve, making it data-selective (shown in Fig. 2). Our experimental comparisons also support our analysis (shown in Tab. 1). To sum up, the LT loss clearly outperforms the GM loss in terms of learning from inaccurate supervision.

4 Experiments

In this section, we first discuss various empirical understandings of our LT losses using CCE and LT-CCE as examples and compare the performance of our approach against noisy labels to other SOTA methods. To discover the influence of the hyperparameters ϵ and λ in the LT loss, some ablation studies are also conducted. Our experiments are supported by six datasets, including MNIST [48], CIFAR-10 [49], CIFAR-100 [49] and three real-world datasets ANIMAL-10N [42], WebVision [50] and ImageNet [3].

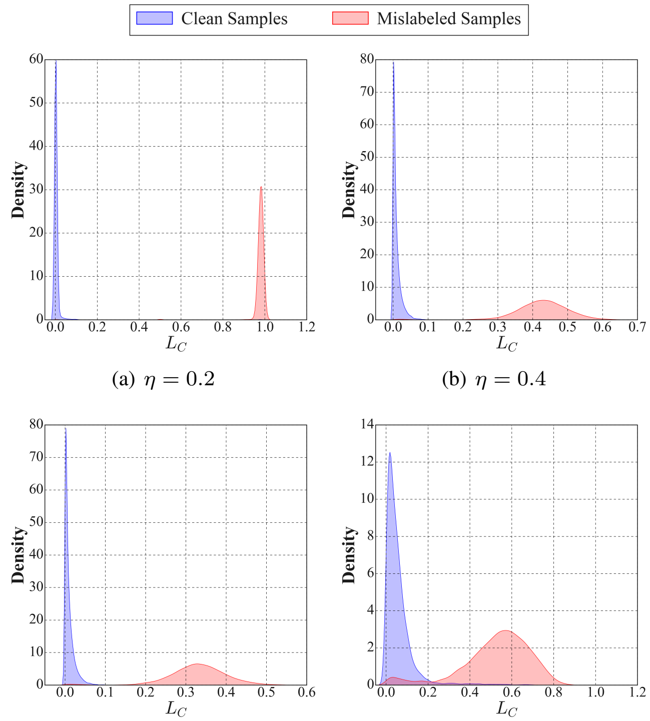
Noise Setting: We analyze both symmetric and asymmetric noise. Symmetric noise is generated by uniformly translating a true label to a random label with probability η, and asymmetric noise is generated using rules that convert a true label to a given label with probability η. In our experiments, we produce asymmetric noise followed [28], [30], [31] in which translating $2 \rightarrow 7$, $3 \rightarrow 8$, $5 \leftrightarrow 6$ and $7 \rightarrow 1$ for MNIST, and TRUCK → AUTOMOBILE, BIRD → AIRPLANE, DEER → HORSE, CAT ↔ DOG for CIFAR-10. As for CIFAR100, we first group the 100 classes into 20 super-classes, with each containing 5 sub-classes, and then translate each class



(a) Training set

(b) Test set

Fig. 5. The accuracy curves during training. (a) shows the accuracy of the training set under various noise rates accompanied by the global step increases, and (b) shows the accuracy of the test set. As can be seen, our strategy can effectively overcome the overfitting caused by the label error.



(a) $\eta = 0.2$

(b) $\eta = 0.4$

(c) $\eta = 0.6$

(d) $\eta = 0.8$

Fig. 6. The densities of clean and mislabeled samples on L_C under various noise rates η of symmetric noise. We can observe that the distribution of clean samples and that of mislabeled samples are different, which reflects our approach can automatically detect even relabel the mislabeled sample by L_C.

within the same super-class into the next in a circular fashion. For the empirical interpretations, symmetric noise with $\eta \in [0.2, 0.8]$ is selected to test. For the robustness evaluation, both symmetric noise with $\eta \in [0.2, 0.8]$ and asymmetric noise with $\eta \in [0.2, 0.4]$ are selected to test.

4.1 Empirical Understandings

Experimental Setup: We build a toy model with two convolutional layers and two fully connected layers and judge some empirical understandings on MNIST. The experiment consists of two parts. First, we explore the feature representation of the LT loss in the penultimate layer. The dimension of the penultimate output is set to two for better visualization. Then, the convergence during the training and the effect of L_C in the LT loss are further evaluated. As such, the dimension of the penultimate output is set to 128. For our LT loss, ϵ and λ are set to 0.3 and 0.05 under $\eta \leqslant 0.6$, while 0.1 and 0.01 under $\eta > 0.6$, respectively. All networks are trained using the Adam optimizer with a learning rate of 0.001, a weight decay of 5×10^{-4}, a batch size of 128, and cosine learning rate annealing. The total epoch is set to 50. The situations in various symmetric noises are picked up for illustration.

More Tolerant and Distinguishable Representation: The feature representation of the training set in the penultimate layer has been shown in Fig. 3. As can be seen, the output features from various categories are dispersed according to their respective projected angles. When using CCE, the clusters are separable and clear under $\eta = 0$, while the areas of different clusters seem to be more imbalanced accompanied by η increases, leading in the omission of some categories under $\eta \geqslant 0.4$ (shown in Figs. 3(a) to 3(e)). Inversely, the harvested representation of LT-CCE is obviously more tolerant and distinguishable, allowing the development of a complete and more acceptable representation even under the extreme noise rates. (shown in Figs. 3(f) to 3(i)).

Strikingly, the effects of our method are convincing for the following two reasons: 1) It brings all-around improvements to various noise rates. We can discover that LT-CCE can obtain more robust clusters than CCE not only under low noise rates but also under high noise rates. 2) It also brings all-around improvements to various categories. When using CCE for training, it is obvious that the influences of the noise label on various categories are distinctive. Some categories are finally missing from the predictions. However, the action of our approach is consistent for different categories and allows them to become more stable.

To better illustrate, we take the noisy category "1" as an example and exhibit the distribution of mislabeled samples in the cluster. As shown in Fig. 4, following CCE training, the mislabeled samples represent obvious overlap with clean samples under $\eta \leqslant 0.4$ (shown in Figs. 4(a) and 4(b)), whereas larger overlap occurs under $\eta > 0.4$ (shown in Figs. 4(c) and 4(d)). Nevertheless, when using LT-CCE for training, nearly all clean samples gather tightly while the mislabeled samples scatter, and there is **little overlap** under all tested noise rates (shown in Figs. 4(e) to 4(h)). These results adequately reveal the validity of our strategy.

As a matter of fact, *since introducing the prior hypothesis of the student distribution in feature representation to resist the chaos of noisy labels, the LT-CCE can produce more tolerant and clearer clusters than the original CCE. It is essential to improve the performance in inaccurate supervision.*

Generalization of Results: It should be illustrated that we only select CCE and LT-CCE as examples to explore the empirical interpretation in this section. However, since clustering samples with the same labels is a common denominator for most popular loss functions (MAE, Focal loss [51], *etc.*), the above interpretations can be generalized to others and their enhanced versions modified by our approach.

More Appropriate Convergence during Training: The accuracy curves of the training/test sets during training are shown in Fig. 5. It can be seen that CCE suffers from a serious overfitting problem. Although the accuracy of the training set can reach a high level (nearly 100%) under all tested noise rates, the accuracy of the test set gradually decreases during convergence. This phenomenon is also mentioned by most of the literature [27], [28], [29], [30] and regarded as one of the main challenges in inaccurate supervision. As for our strategy, we observe that although the accuracy of LT-CCE is lower in the training set than that of CCE, it can reach a high level in the test set in all tested situations. Additionally, the training accuracy almost coincides with the rate of clean samples in the noisy cluster $(1-\eta)$ shown in Fig. 5(a). This result reveals that the contributions of mislabeled samples towards the convergence can be little. As deduced in Sec. 3.3, our method diminishes the weights of mislabeled samples in the gradient by introducing the prior probabilistic assumption. The long-tail characteristic of the student distribution is applied to naturally resist the convergence of mislabeled samples. Thanks to this reason, the performance of LT-CCE is much greater than that of CCE at all tested noise rates (shown in Fig. 5(b)).

In other words, *we not only theoretically but also empirically demonstrate that LT losses have strong convergence abilities on clean samples but weaken on mislabeled samples. It reflects the outstanding data-selective characteristic of our approach, which is especially significant in inaccurate supervision.*

Different Distribution Patterns on L_c: In Sec. 3.4, we illustrate that the incorrect label can be detected by our LT loss. In our experiments, we further explore the densities of clean samples and mislabeled samples on L_C and discover that the distribution of clean samples and that of mislabeled samples are totally different. As shown in Fig. 6, the mislabeled sample represents a larger L_C than the clean sample. The distance between the two distributions is far under the low noise rate of 0.2, while there are clear boundaries in all tested situations. Apparently, mislabeled samples are close to the clusters of their natural categories but far away from others with our LT losses, which provides an opportunity to automatically identify and even relabel them according to L_C. Attaching it makes our LT losses have adaptive amendment ability. Unluckily, since relabeling samples based on L_C should consider tremendous rules that are not our focus, the reported results in this paper are not subject to label correction. It will be left to our future work to design further. Additionally, it should be illustrated that since the density curves in Fig. 6 are obtained by filtering with the kernel function, they have some responses in the negative axis of L_C. The values of L_C are non-negative in all experiments.

4.2 Robustness Evaluation with Other Robust Losses

Baseline: We compare our LT loss with five SOTA robust losses as well as the CCE loss: (1) GCE [28]: a training loss by combining MAE and CCE losses; (2) SCE [30]: a training loss by combining RCE and CCE losses inspired by the symmetricity of the Kullback-Leibler divergence; (3) APL [31]: a training loss by combining active losses with one passive loss. We select the best-reported combination, Normalized Cross-Entropy (NCE) + RCE, as our baseline; (4) JNPL [33]: a training loss regarded as an enhanced version of NLNL [29]; (5) JS [34]: a training loss adopting the generalized Jensen-Shannon divergence for multiple distributions.

Experimental Setup: We attempt to observe the variations when the baselines are strengthened by our method on MNIST, CIFAR10 and CIFAR100 datasets. Since the reported results of baselines in the literature are generated in different experimental environments (different models, different noise settings, *etc.*), we reproduce them in our experiments for fairness in comparison. The hyperparameter settings and codes are derived from their literature and released programs. Experiments are conducted with a two-layer CNN for MNIST, a six-layer CNN for CIFAR10 (used in the experiments of empirical understandings) and the ResNet34 for CIFAR100, and the epoch is set to 50, 120, and 200, respectively. All networks are trained using the Adam optimizer with a learning rate of 0.001, a weight decay of 5×10^{-4}, a batch size of 128, a gradient clip of five, and cosine learning rate annealing in all experiments. For MNIST, ϵ and λ are set to 0.3 and 0.05 under $\eta\in[0.2,0.6]$, while 0.1 and 0.01 under $\eta=0.8$ in symmetric noise experiments, respectively. They are set to 0.3 and 0.05 in the asymmetric noise experiments, respectively. For CIFAR10, ϵ and λ are set to 0.1 and 0.05 under $\eta\in[0.2,0.6]$, while 0.01 and 0.001 under $\eta=0.8$ in symmetric noise experiments, respectively. They are set to 0.1 and 0.05 in the asymmetric noise experiments, respectively. For CIFAR100, ϵ and λ are set to 0.05 and 0.05 under $\eta\in[0.2,0.4]$, while 0.01 and 0.001 under $\eta\in[0.6,0.8]$ in symmetric noise experiments, respectively. They are set to 0.01 and 0.005 when using CCE, GCE, SCE and JNPL for training, and to 0.05 and 0.05 when using APL and JS for training in the asymmetric noise experiments, respectively.

Robustness Performance: The classification accuracy is reported in Tab. 2. As can be seen, the LT losses outperform the baselines under most of situations, and various improvements are larger than 20%. On MNIST, the largest gap under symmetric noise represents 62.79% (85.80% - 23.01%) appearing when employing CCE and LT-CCE with $\eta=0.8$, while that under asymmetric noise represents 10.83% (96.52% - 85.69%) appearing when employing SCE and LT-SCE with $\eta=0.4$. On CIFAR10, the largest gap under symmetric noise represents 37.87% (79.76% - 41.89%) appearing when employing CCE and LT-CCE with $\eta=0.6$, while that under asymmetric noise represents 5.64% (86.72% - 81.08%) appearing when employing SCE and LT-SCE with $\eta=0.4$. On CIFAR100, the largest gap under symmetric noise represents 23.31% (46.02% - 22.71%) appearing when employing CCE and LT-CCE with $\eta=0.6$, while that under asymmetric noise represents 13.39% (62.59% - 49.20%) appearing when employing APL and LT-APL with $\eta=0.2$. These results adequately demonstrate the validity of our approach. Furthermore, it can be observed that the effect of the LT loss under the symmetric noise is on average better than that under the asymmetric noise, and even negative influences are generated under the asymmetric noise in a few cases (such as some experiments on CIFAR100). We guess that the randomness in symmetric noise is more in line with the unbiased characteristic of the student distribution, which makes our approach better for handling the symmetric noise. The specific reasons and solutions for the degeneration will be further explored and improved in the future.

Besides, recent advances in explainable deep learning push the development of feature visualization for classification decisions. We attempt to employ the Grad-CAM [52] for exploring the differences in feature extraction between the baseline and our method (using CCE and LT-CCE as an example) under various noise rates on CIFAR10. The experimental conditions are as above, and the visualized results are shown in Fig. 7. As can be seen, the label noise undoubtedly affects the accuracy of feature extraction, not only for the baseline but also for our method. Nevertheless, the influence on our approach is much less. It is obvious that the areas of interest for the baseline are mostly **incorrect and unstable** under the noisy labels, and this phenomenon seems to be more apparent under high noise rates,

TABLE 2
Test accuracy (%) of different methods on benchmark datasets under various rates of symmetric and asymmetric noise. The average accuracy and the standard deviation of three random runs are reported, and the best paired results are in **bold**.

Datasets (Architecture)	Methods	Symmetric Noise				Asymmetric Noise		
		Noise Rate η				Noise Rate η		
		0.2	0.4	0.6	0.8	0.2	0.3	0.4
MNIST (Two-layer CNN)	CCE	90.50 ± 0.20	73.48 ± 0.36	50.13 ± 0.80	23.01 ± 0.58	93.02 ± 0.39	87.24 ± 0.10	81.70 ± 0.15
	LT-CCE	**98.92 ± 0.03**	**98.75 ± 0.10**	**97.76 ± 0.17**	**85.80 ± 0.71**	**98.83 ± 0.08**	**96.78 ± 0.08**	**89.18 ± 0.06**
	GCE [28]	98.93 ± 0.02	97.64 ± 0.11	90.85 ± 0.19	63.51 ± 3.57	97.19 ± 0.18	91.49 ± 0.45	83.23 ± 0.57
	LT-GCE	**98.96 ± 0.01**	**98.76 ± 0.08**	**97.59 ± 0.07**	**84.62 ± 1.36**	**98.76 ± 0.03**	**96.31 ± 0.27**	**89.17 ± 0.40**
	SCE [30]	**98.96 ± 0.10**	97.63 ± 0.21	89.10 ± 0.79	54.53 ± 1.62	98.24 ± 0.07	94.39 ± 0.40	85.69 ± 0.38
	LT-SCE	98.81 ± 0.11	**98.78 ± 0.07**	**98.21 ± 0.18**	**93.80 ± 0.50**	**98.96 ± 0.05**	**98.51 ± 0.07**	**96.52 ± 0.24**
	APL [31]	**99.06 ± 0.07**	98.01 ± 0.14	92.85 ± 0.10	56.37 ± 0.45	98.59 ± 0.07	96.03 ± 0.13	88.08 ± 0.39
	LT-APL	99.04 ± 0.10	**98.78 ± 0.04**	**98.08 ± 0.10**	**85.70 ± 0.41**	**99.03 ± 0.03**	**98.41 ± 0.10**	**96.16 ± 0.41**
	JNPL [33]	97.77 ± 0.03	91.24 ± 0.69	64.55 ± 0.46	27.96 ± 0.24	97.53 ± 0.10	89.19 ± 0.38	80.87 ± 0.62
	LT-JNPL	**98.88 ± 0.11**	**98.30 ± 0.03**	**91.05 ± 1.15**	**50.26 ± 2.94**	**97.92 ± 0.16**	**93.15 ± 0.20**	**83.49 ± 1.24**
	JS [34]	**98.89 ± 0.06**	96.95 ± 0.33	93.43 ± 0.06	63.00 ± 1.72	93.07 ± 0.32	87.40 ± 0.44	81.55 ± 0.13
	LT-JS	98.84 ± 0.10	**98.84 ± 0.01**	**97.76 ± 0.16**	**87.99 ± 0.74**	**98.86 ± 0.07**	**97.25 ± 0.11**	**90.07 ± 0.67**
CIFAR10 (Six-layer CNN)	CCE	77.53 ± 0.50	58.95 ± 0.56	41.89 ± 0.57	42.35 ± 2.42	85.39 ± 0.51	81.25 ± 0.12	76.35 ± 0.43
	LT-CCE	**89.92 ± 0.24**	**86.54 ± 0.30**	**79.76 ± 0.24**	**47.84 ± 0.91**	**88.53 ± 0.32**	**83.60 ± 0.43**	**76.66 ± 0.32**
	GCE [28]	89.43 ± 0.06	84.18 ± 0.13	71.75 ± 0.36	48.98 ± 1.06	87.58 ± 0.30	81.50 ± 0.14	75.38 ± 0.17
	LT-GCE	**90.01 ± 0.09**	**86.72 ± 0.22**	**79.89 ± 0.17**	**50.43 ± 0.82**	**88.68 ± 0.08**	**83.88 ± 0.51**	**77.10 ± 0.10**
	SCE [30]	87.31 ± 0.15	78.10 ± 0.10	57.76 ± 0.41	37.14 ± 1.10	86.37 ± 0.39	81.08 ± 0.29	75.63 ± 0.16
	LT-SCE	**89.38 ± 0.04**	**86.72 ± 0.09**	**80.68 ± 0.49**	**39.96 ± 1.70**	**89.27 ± 0.20**	**86.72 ± 0.39**	**80.38 ± 0.51**
	APL [31]	**89.64 ± 0.28**	86.63 ± 0.06	80.65 ± 0.11	39.97 ± 3.66	89.13 ± 0.18	85.66 ± 0.18	79.24 ± 0.46
	LT-APL	89.42 ± 0.13	**86.82 ± 0.18**	**80.93 ± 0.30**	**40.87 ± 1.57**	**89.28 ± 0.24**	**86.29 ± 0.36**	**79.99 ± 0.58**
	JNPL [33]	83.09 ± 0.14	74.18 ± 0.58	53.47 ± 0.47	36.51 ± 2.19	**87.35 ± 0.21**	**83.12 ± 0.70**	**77.29 ± 0.10**
	LT-JNPL	**87.62 ± 0.19**	**83.46 ± 0.81**	**74.44 ± 1.03**	**40.57 ± 2.68**	87.17 ± 0.14	82.07 ± 0.23	76.00 ± 0.18
	JS [34]	89.41 ± 0.10	83.45 ± 0.34	76.21 ± 0.13	**52.01 ± 1.90**	84.76 ± 0.39	80.63 ± 0.36	75.05 ± 0.36
	LT-JS	**89.93 ± 0.05**	**86.73 ± 0.31**	**80.70 ± 0.25**	51.02 ± 1.19	**88.94 ± 0.14**	**84.99 ± 0.18**	**77.39 ± 0.16**
CIFAR100 (ResNet34)	CCE	56.87 ± 0.76	40.38 ± 1.37	22.71 ± 1.45	7.51 ± 0.40	58.11 ± 0.46	50.65 ± 0.63	**42.26 ± 0.52**
	LT-CCE	**65.91 ± 0.82**	**58.69 ± 0.93**	**46.02 ± 0.98**	**26.83 ± 0.18**	**62.37 ± 0.47**	**52.47 ± 0.89**	41.02 ± 0.39
	GCE [28]	64.99 ± 1.03	56.88 ± 0.62	45.51 ± 1.42	26.12 ± 0.50	**64.24 ± 1.03**	**57.26 ± 0.57**	**44.08 ± 1.48**
	LT-GCE	**66.02 ± 0.74**	**58.61 ± 1.31**	**46.15 ± 1.09**	**26.88 ± 0.80**	61.35 ± 0.51	52.41 ± 0.50	41.01 ± 0.38
	SCE [30]	55.73 ± 1.40	39.30 ± 0.84	21.68 ± 1.43	6.90 ± 0.18	**58.03 ± 0.42**	**49.32 ± 0.68**	**41.08 ± 0.42**
	LT-SCE	**63.75 ± 0.34**	**55.37 ± 0.70**	**43.37 ± 1.24**	**24.07 ± 0.09**	55.58 ± 0.39	45.54 ± 0.62	37.08 ± 0.23
	APL [31]	51.06 ± 2.87	41.13 ± 2.65	28.87 ± 1.01	9.57 ± 0.56	49.20 ± 3.89	43.95 ± 2.05	31.66 ± 2.28
	LT-APL	**63.29 ± 0.49**	**54.70 ± 1.73**	**40.52 ± 1.65**	**22.63 ± 0.78**	**62.59 ± 1.31**	**56.90 ± 1.29**	**44.05 ± 1.32**
	JNPL [33]	54.90 ± 0.52	37.96 ± 1.38	21.17 ± 2.17	6.92 ± 0.22	**58.18 ± 1.19**	**49.61 ± 0.62**	**40.00 ± 1.50**
	LT-JNPL	**61.44 ± 0.52**	**52.93 ± 0.07**	**35.58 ± 0.95**	**20.20 ± 1.00**	54.98 ± 0.02	44.96 ± 0.46	36.40 ± 0.58
	JS [34]	64.21 ± 0.85	56.24 ± 0.31	**43.26 ± 1.62**	22.42 ± 0.52	63.02 ± 0.68	53.73 ± 0.31	40.62 ± 0.85
	LT-JS	**64.40 ± 0.42**	**57.45 ± 2.61**	40.43 ± 0.98	**23.82 ± 0.92**	**64.48 ± 0.39**	**57.58 ± 0.48**	**43.26 ± 0.05**

while the extracted features obtained by our method are relatively **exact and stable** under various noise rates. These results also illustrate the advancement of our approach from the perspective of explainable feature extraction.

Generally speaking, *the LT loss family produce better performance than their original SOTA versions under most of tasks, which illustrates the generalization and advancement of our approach in inaccurate supervision.*

Next, we evaluate the performance of our LT losses on some real-world noisy datasets. Specifically, ANIMAL-10N, Webvision, and ImageNet ILSVRC12 are applied to explore. ANIMAL-10N [42] contains 10 animals with confusing appearances. The estimated label noise rate is around 8%. There are 50,000 training images and 5,000 testing images. Webvision [50] contains 2.4 million images. It has 1,000 categories, the same as the ImageNet ILSVRC12. The estimated label noise rate is around 20%. Similar to [31], [53], the first 50 categories of the Google image subset are selected as the training data, and we evaluate on both WebVision and ILSVRC12 validation set. In our experiments, 19 SOTA approaches to deal with learning with noisy labels [18], [19], [20], [28], [30], [31], [35], [42], [53], [54], [55], [56], [56], [57], [58], [59], [60] as well as CCE are employed as our baselines for comparisons.

Experimental Setup: We employ the CCE, GCE and SCE as L_{tds} in the experiments. When using GCE, we set q to 0.001. When using SCE, we set $\alpha = 6$ and $\beta = 0.1$. For ANIMAL-10N, VGG19-BN backbone is applied for training. The batch size and the epoch are set to 128 and 200, respectively. For WebVision, Inception-ResNet backbone is utilized. The batch size and the epoch are set to 32 and 200, respectively. All networks are trained using the Stochastic Gradient Descent (SGD) optimizer with cosine learning rate annealing. The weight decay is set to 1×10^{-3} for ANIMAL-10N while 5×10^{-4} for WebVision. The learning rate is set to 0.1 for ANIMAL-10N and 0.01 for WebVision, respectively. Additionally, the Random Crop, Random Horizontal Flip, and CutMix are picked as data augmentation strategies. ϵ and λ are set to 0.01 and 0.05 in all experiments.

4.3 Experiments on Real-World Noisy Dataset

Results: The classification accuracy on real-world datasets is reported in Tabs. 3 and 4. As can be seen, compared to the original versions of CCE, GCE and SCE, LT-CCE, LT-GCE, and LT-SCE obtain much greater performance. The gap between CCE and LT-CCE on ANIMAL-10N is 5.90% (85.30%-79.4%). The gaps in top-1 accuracy between CCE, GCE, and SCE and their enhanced versions on ILSVRC12 are 14.12% (73.00%-58.88%), 17.72% (71.40%-53.68%) and 11.92% (73.68%-61.76%), respectively. Moreover, our methods outperform other SOTA strategies in most cases. Except from the top-1 accuracy on ILSVRC12, all LT losses yield the best result in our experiments. Meanwhile, in the tested LT loss family, we discover that LT-SCE seems to be more effective in real-world situations.

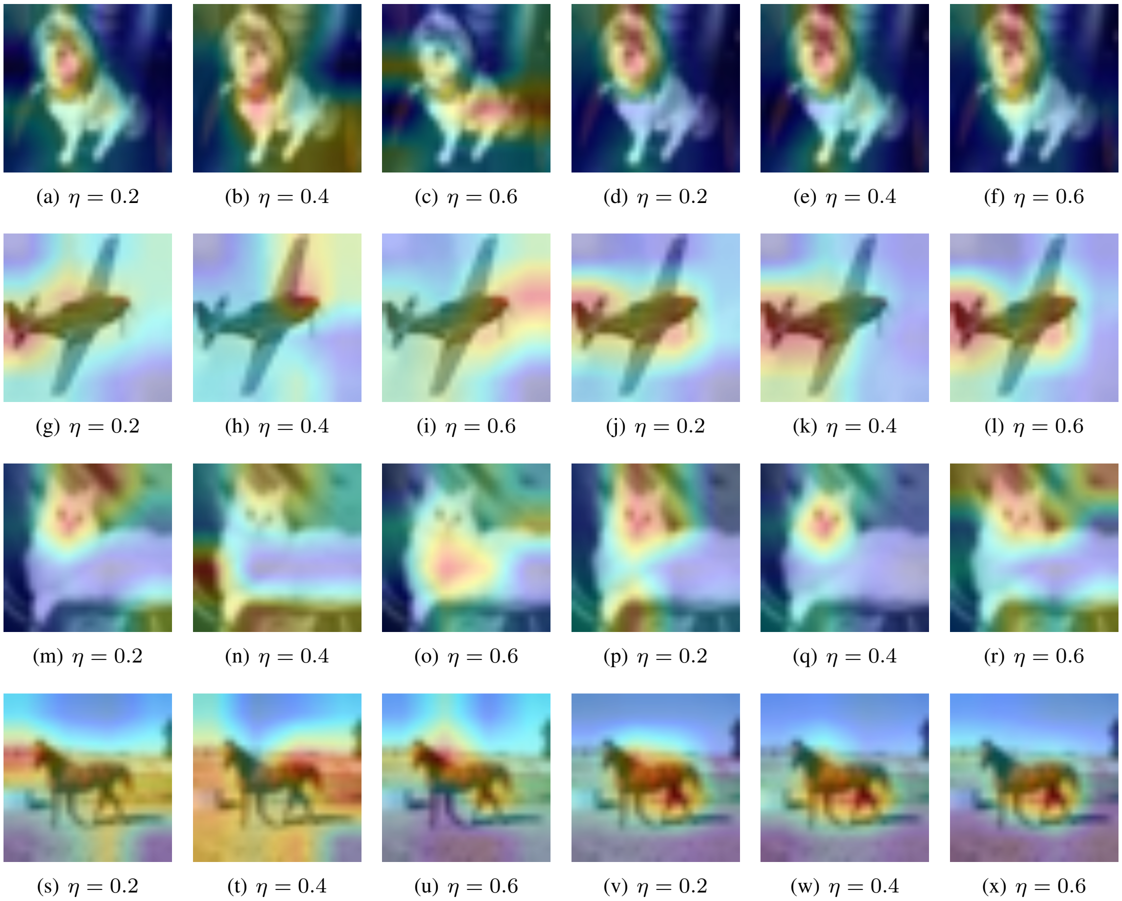


Fig. 7. The feature visualiations of model predictions using Grad-CAM on CIFAR10. The red/blue areas have larger/smaller weights for the predictions. The first three images in each row are from the baseline, and the last three are from our approach. It is obvious that our LT loss can obtain more robust and exact features than the baseline under various η.

Additionally, it should be highlighted that our strategy is not mutually exclusive with the baseline methods. In other words, it is convenient to employ our method with other SOTA methods for better performance. In fact, the generality is regarded as the essential advantage of our approach and has been supported by the earlier paragraph (see Tab. 2). Generally speaking, these results demonstrate that our LT losses are still resistant to label noise from the real world and can be a competitive solution compared to other SOTA approaches.

4.4 Ablation Studies

Finally, to further explore the influence of hyperparameter settings, some ablation studies are conducted. We train the model using LT-CCE on CIFAR10 and take it as an example to illustrate.

Experimental Setup: Specifically, we apply the same structure in Sec. 4.2 (a six-layer CNN) and attempt to observe the variances under a low noise rate of 0.2 and a high noise rate of 0.6. When changing ϵ, we set λ to 0.05, and the ϵ is set to [0, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5]. When changing λ, we set ϵ to 0.1, and the λ is set to [0, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5]. All networks are trained using the Adam optimizer with a learning rate of 0.001, a 5×10^{-4} weight decay, a batch size of 128 and a gradient clip of five in all experiments.

Influences of Hyperparameters: As shown in Figs. 8(a) and 8(c), we observe that setting ϵ to a big value represents an underfitting phenomenon, and it is more obvious under a high noise rate of 0.6. In fact, employing the metric learning strategy increases the difficulty of classification, especially under inaccurate supervision. However, it can improve performance with a suitable setting under conditions of high noise. We discover that there is a slight overfitting under $\eta = 0.6$ with setting $\epsilon = 0$, while setting it to a small value successfully fights against the degradation. Of note, the overfitting problem is common in inaccurate supervision and become acuter accompanied by η increases [30], [31]. Introducing metric learning with a small weight can effectively restrain the overfitting problem under conditions of high noise rates. As for λ shown in Figs. 8(b) and 8(d), we observe that L_C can improve the speed of convergence, which is consistent with our description in Sec. 3.4. However, the overfitting problem is also exposed while setting λ to a large value. To sum up, we recommend setting ϵ and λ to a relatively small value according to various noise rates.

5 Limitations

Limitations exist in the current study. Firstly, the L_C in our LT loss can be applied to detect and even relabel the mislabeled sample as shown in Sec. 4.1, implying the potential error correction capability of our method. We speculate that the performance can be further improved by excluding or relabeling the mislabeled samples with L_C, but it is not currently explored yet. Secondly, we observe in Sec. 4.2 that our approach can enhance various SOTA losses in most cases but is not ideal in a few cases. The reasons

TABLE 3
Test accuracy (%) of different methods on ANIMAL-10N datasets. The best result is in **bold**.

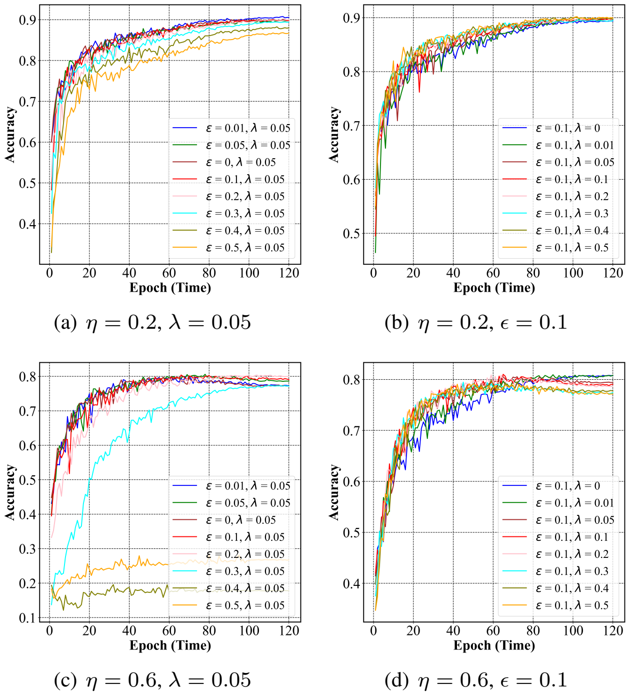
Methods	CCE	Nested [54]	CED [54]	SELEIE [42]	PLC [55]	NCT [54]	LT-CCE	LT-GCE	LT-SCE
Accuracy	79.4⋆	81.3	81.3	81.8	83.4	84.1	85.30	85.30	**85.66**

⋆: Results reported in [55].

TABLE 4
Test accuracy (%) of different methods on WebVision and ILSVRC12 datasets. The best two results are in **bold**.

Methods	WebVision		ILSVRC12	
	top-1	top-5	top-1	top-5
CCE	-	-	58.88⋆	-
GCE [28]	-	-	53.68⋆	-
Forward [35]	61.12	82.68	57.36	82.36
Decoupling [19]	62.54	84.74	58.26	82.26
D2L [41]	62.68	84.00	57.80	81.36
SCE [30]	-	-	61.76⋆	-
APL [31]	-	-	62.64	-
MentorNet [18]	63.00	81.40	57.80	79.92
Co-teaching [20]	63.58	85.20	61.48	84.98
Iterative-CV [58]	65.24	85.34	61.60	84.98
DivideMix [56]	77.32	91.26	**75.20**	90.84
ELR+ [57]	77.78	91.68	70.29	89.76
SPR [53]	78.12	-	-	-
ProtoMix [59]	76.3	91.5	73.3	91.2
MoPro [60]	77.59	-	**76.31**	-
LT-CCE	77.68	92.16	72.96	91.56
LT-GCE	**78.72**	**92.40**	73.32	**91.68**
LT-SCE	**79.04**	**93.12**	73.44	**91.80**

⋆: Results reported in [31].



(a) $\eta = 0.2, \lambda = 0.05$ (b) $\eta = 0.2, \epsilon = 0.1$

(c) $\eta = 0.6, \lambda = 0.05$ (d) $\eta = 0.6, \epsilon = 0.1$

Fig. 8. The accuracy curves on the test set of CIFAR10 during training under various noise rates η of symmetric noise with different hyperparameter settings.

and solutions are currently ambiguous. Thirdly, since the essence of the LT loss is to tolerate intra-class discrepancies, our approach may work not only for the noisy label, but also for the noisy data. Unfortunately, the present work does not cover it, either. These three key issues are left as future improvements to our LT loss.

6 Conclusions

This study set out to improve the robustness of deep learning with noisy labels and indicated, not only empirically but also theoretically, that assuming identically labeled deep features to follow the student distribution could yield promising performance. The analysis of the feature representation undertaken here has extended the knowledge of existing robust losses and allowed us to create a family of new losses called student losses. Strikingly, the sharp shift in the probability distribution made the student loss naturally data-selective, and various losses could be strengthened to be student losses. After that, we further introduced some metric learning strategies and developed the LT loss. Experiments on both benchmark and real-world datasets demonstrated that the LT loss outperformed the baseline. Overall, we believe the LT loss is a up-and-coming perspective in inaccurate supervision and will become a popular technique to deal with noisy labels.

Acknowledgments

The authors would like to thank Dr. Xing-Yao Wang and Dr. Hong-Xiang Gao from Southeast University in Nanjing, China, for participating in discussions of this manuscript. The authors also thank Dr. David Hike from Harvard Medical School in Boston, America, for editing the language of this manuscript.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255. 1

[2] C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," *arXiv:2103.14749*, 2021. 1

[3] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, 2017. 1, 6

[4] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 1431–1439. 1, 2

[5] S. Sukhbaatar, J. Bruna, M. Paluri, L. Bourdev, and R. Fergus, "Training convolutional networks with noisy labels," in *Proc. Int. Conf. Learn. Represent. Worksh.*, 2015, pp. 1–11. 1, 2

[6] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 44–53, 2014. 1, 2

[7] J. Goldberger and E. Ben-Reuven, "Training deep neural-networks using a noise adaptation layer," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–9. 1, 2

[8] A. J. Bekker and J. Goldberger, "Training deep neural-networks based on unreliable labels," in *Proc. ICASSP*, 2016, pp. 2682–2686. 1, 2

[9] B. Han *et al.*, "Masking: A new perspective of noisy supervision," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 5836–5846. 1, 2

[10] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 2691–2699. 1, 2

[11] J. Yao *et al.*, "Deep learning from noisy image labels with quality embedding," *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1909–1922, 2018. 1, 2

[12] S. Jenni and P. Favaro, "Deep bilevel learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 618–633. 1, 3

[13] R. Tanno, A. Saeedi, S. Sankaranarayanan, D. C. Alexander, and N. Silberman, "Learning from noisy labels by regularized estimation of annotator confusion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 11 244–11 253. 1, 3

[14] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *Proc. ICML*, 2019, pp. 2712–2721. 1, 3

[15] X. Xia *et al.*, "Robust early-learning: Hindering the memorization of noisy labels," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–15. 1, 3

[16] A. K. Menon, A. S. Rawat, S. J. Reddi, and S. Kumar, "Can gradient clipping mitigate label noise?" in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–26. 1, 3

[17] H. Wei, L. Tao, R. Xie, and B. An, "Open-set label noise can improve robustness against inherent label noise," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–15. 1, 3

[18] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Proc. ICML*, 2018, pp. 2304–2313. 2, 3, 9, 11

[19] E. Malach and S. Shalev-Shwartz, "Decoupling 'when to update' from 'how to update'," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 960–970. 2, 3, 9, 11

[20] B. Han *et al.*, "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8527–8537. 2, 3, 9, 11

[21] X. Yu, B. Han, J. Yao, G. Niu, I. W. Tsang, and M. Sugiyama, "How does disagreement help generalization against label corruption?" in *Proc. ICML*, 2019, pp. 7164–7173. 2, 3

[22] P. Chen, B. Liao, G. Chen, and S. Zhang, "Understanding and utilizing deep neural networks trained with noisy labels," in *Proc. ICML*, 2019, pp. 1062–1070. 2, 3

[23] H. Wei, L. Feng, X. Chen, and B. An, "Combating noisy labels by agreement: A joint training method with co-regularization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 13 726–13 735. 2, 3

[24] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–14. 2, 3

[25] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 5050–5060. 2, 3

[26] T. Zhou, S. Wang, and J. Bilmes, "Robust curriculum learning: From clean label detection to noisy label self-correction," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–18. 2, 3

[27] A. Ghosh, H. Kumar, and P. Sastry, "Robust loss functions under label noise for deep neural networks," in *Proc. AAAI*, 2017, pp. 1919–1925. 2, 8

[28] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 8778–8788. 2, 6, 8, 9, 11

[29] Y. Kim, J. Yim, J. Yun, and J. Kim, "Nlnl: Negative learning for noisy labels," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 101–110. 2, 8

[30] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey, "Symmetric cross entropy for robust learning with noisy labels," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 322–330. 2, 6, 8, 9, 10, 11

[31] X. Ma, H. Huang, Y. Wang, S. Romano, S. Erfani, and J. Bailey, "Normalized loss functions for deep learning with noisy labels," in *Proc. ICML*, 2020, pp. 6543–6553. 2, 6, 8, 9, 10, 11

[32] Y. Lyu and I. W. Tsang, "Curriculum loss: Robust learning and generalization against label corruption," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–22. 2

[33] Y. Kim, J. Yun, H. Shon, and J. Kim, "Joint negative and positive learning for noisy labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 9442–9451. 2, 8, 9

[34] E. Englesson and H. Azizpour, "Generalized jensen-shannon divergence loss for learning with noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, pp. 30 284–30 297. 2, 8, 9

[35] G. Patrini, A. Rozza, A. Krishna Menon, R. Nock, and L. Qu, "Making deep neural networks robust to label noise: A loss correction approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 1944–1952. 2, 3, 9, 11

[36] X. Xia *et al.*, "Are anchor points really indispensable in label-noise learning?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–12. 2, 3

[37] Y. Yao *et al.*, "Dual t: Reducing estimation error for transition matrix in label-noise learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 7260–7271. 2, 3

[38] H. S. Chang, E. Learned-Miller, and A. McCallum, "Active bias: Training more accurate neural networks by emphasizing high variance samples," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1002–1012. 2, 3

[39] H. Zhang, X. Xing, and L. Liu, "Dualgraph: A graph-based method for reasoning about label noise," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2021, pp. 9654–9663. 2, 3

[40] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–11. 2, 3

[41] X. Ma *et al.*, "Dimensionality-driven learning with noisy labels," in *Proc. ICML*, 2018, pp. 3355–3364. 2, 3, 11

[42] H. Song, M. Kim, and J.-G. Lee, "Selfie: Refurbishing unclean samples for robust deep learning," in *Proc. ICML*, 2019, pp. 5907–5915. 2, 3, 6, 9, 11

[43] W. Wan, Y. Zhong, T. Li, and J. Chen, "Rethinking feature distribution for loss functions in image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 9117–9126. 4, 5

[44] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 499–515. 4

[45] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. ICML*, 2016, pp. 507–516. 4

[46] S. Sun, W. Chen, L. Wang, X. Liu, and T.-Y. Liu, "On the depth of deep neural networks: A theoretical view," in *Proc. AAAI*, 2016, pp. 2066–2072. 4

[47] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, "Early-learning regularization prevents memorization of noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 20 331–20 342. 4

[48] Y. LeCun, C. Cortes, and C. J. Burges., *The MNIST Database of Handwritten Digits*, 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist 6

[49] A. Krizhevsky, V. Nair, and G. Hinton, *CIFAR-10 and CIFAR-100 Datasets*, 2014. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html 6

[50] W. Li, L. Wang, W. Li, E. Agustsson, and L. V. Gool, "Web vision database: Visual learning and understanding from web data," *arXiv:1802.05300*, 2017. 6, 9

[51] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 2980–2988. 7

[52] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proc. Int. Conf. Comput. Vis.*, 2017, pp. 618–626. 8

[53] Y. Wang, X. Sun, and Y. Fu, "Scalable penalized regression for noise detection in learning with noisy labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2022, pp. 346–355. 9, 11

[54] Y. Chen, X. Shen, S. X. Hu, and J. A. K. Suykens, "Boosting co-teaching with compression regularization for label noise," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2021, pp. 2688–2692. 9, 11

[55] Y. Zhang, S. Zheng, P. Wu, M. Goswami, and C. Chen, "Learning with feature-dependent label noise: A progressive approach," in *Proc. Int. Conf. Learn. Represent.*, 2021. 9, 11

[56] J. Li, R. Socher, and S. C. Hoi, "Dividemix: Learning with noisy labels as semi-supervised learning," in *Proc. Int. Conf. Learn. Represent.*, 2020. 9, 11

[57] S. Liu, J. Niles-Weed, N. Razavian, and C. Fernandez-Granda, "Early-learning regularization prevents memorization of noisy labels," in *Proc. Adv. Neural Inf. Process. Syst.*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020, pp. 20 331–20 342. 9, 11

[58] P. Chen, B. B. Liao, G. Chen, and S. Zhang, "Understanding and utilizing deep neural networks trained with noisy labels," in *Proc. ICML*, 2019, pp. 1062–1070. 9, 11

[59] J. Li, C. Xiong, and S. Hoi, "Learning from noisy data with robust representation learning," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 9485–9494. 9, 11

[60] J. Li, C. Xiong, and S. Hoi, "Mopro: Webly supervised learning with momentum prototypes," in *Proc. Int. Conf. Learn. Represent.*, 2021. 9, 11

Shuo Zhang received the BS degree in biomedical engineering from Southern Medical University, China, and the MS degree in biomedical engineering from China Medical University, China, in 2017 and 2020, respectively. He is currently working toward the PhD degree with the School of Biological Science and Medical Engineering, Southeast University, China. His main research interests include machine learning and inaccurate supervision, especially in learning from health data.

Jian-Qing Li received the Ph.D. degree in Measurement Technology and Instruments in Southeast University. He is currently a Professor and Vice Present at the School of Basic Medical Sciences, Nanjing Medical University, China. He is also a Professor with the School of Instrument Science and Engineering, Southeast University, China. His research topics include: mHealth and wireless network.

Hamido Fujita (Life senior member, IEEE) received the master's and Ph.D. degrees in information engineering from Tohoku University, Sendai, Japan, in 1985 and 1988, respectively, and the Doctor Honoris Causa from Óbuda University, Budapest, Hungary, in 2013, and Timisoara Technical University, Timisoara, Romania, in 2018. He is a recipient of title of Honorary Professor from Óbuda University in 2011 and the Honorary Scholar Award from the University of Technology Sydney, in 2012. He is also a Highly Cited Researcher in Crossfield for the year 2019 and in computer science for the year 2020, 2021, and 2022 respectively by Clarivate Analytics. He is currently a Distinguished Professor of artificial intelligence with Iwate Prefectural University, Takizawa, Japan. He was an Adjunct Professor of computer science and artificial intelligence with Stockholm University, Stockholm, Sweden; the University of Technology Sydney, Ultimo, NSW, Australia; National Taiwan Ocean University, Keelung, Taiwan; National Taipei University of Technology. He is Research Professor at the University of Granada, Granada, Spain, HTECH University, Vietnam, Harbin Engineering University China, Malaysia-Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia and others. He is Chairman of the i-SOMET Incorporated Association. He has supervised Ph.D. students jointly with the University of Laval, Quebec City, QC, Canada; the University of Technology Sydney; Oregon State University, Corvallis, OR, USA; the University of Paris 1 Pantheon-Sorbonne, Paris, France; and many others. He has given many keynotes in many prestigious international Conferences on intelligent system and subjective intelligence. He headed a number of projects including intelligent HCI, a project related to mental cloning for healthcare system as an intelligent user interface between human users and computers, and SCOPE project on virtual doctor systems for medical applications. He was the Editor-in-Chief for Knowledge-Based Systems (2005 to 2020), and he is now the Emeritus Editor of Knowledge-Based Systems. He is currently the Editor-in-Chief for Applied Intelligence (Springer), and Editor-in-Chief of Healthcare Management (Tayler&Francis).

Yu-Wen Li received the Ph.D. degree from Xiamen University, Xiamen, China, in 2019. Currently, she is a Lecturer with the School of Instrument Science and Engineering, Southeast University, Nanjing, China. Her main research topics include biomedical signal processing, ECG, real-time monitoring, and health big data processing.

Deng-Bao Wang received the BSc degree in computer science from Yantai University, China, and the MSc degree in computer science from Southwest University, China, in 2016 and 2019, respectively. He is currently working toward the PhD degree with the School of Computer Science and Engineering, Southeast University, China. His main research interests include machine learning and data mining, especially in learning from weakly supervised data.

Ting-Ting Zhu received the D.Phil. degree in information and biomedical engineering from the Institute of Biomedical Engineering, Oxford University, Oxford, U.K., in 2016. She is currently a Royal Academy of Engineering Research Fellow and a member of faculty with the Department of Engineering Science, University of Oxford. Her research interests lie in machine learning for healthcare applications. Her work involves the development of machine learning for understanding complex patient data, with an emphasis on Bayesian inference, deep learning, and applications involving low-income countries.

Min-Ling Zhang (Senior member, IEEE) received the BSc, MSc, and the PhD degrees in computer science from Nanjing University, China, in 2001, 2004, and 2007, respectively. He is currently a professor with the School of Computer Science and Engineering, Southeast University, China. His main research interests include machine learning and data mining. He was the general co-chairs of ACML 2018, program co-chairs of CCDM 2020, PAKDD 2019, CCF-ICAI 2019, ACML 2017, CCFAI 2017, and PRICAI 2016, a senior PC member or the area chair of KDD 2021, AAAI 2017-2020, IJCAI 2017–2022, and ICDM 2015–2021. He is on the editorial board of the IEEE Transactions on Pattern Analysis and Machine Intelligence, ACM Transactions on Intelligent Systems and Technology, Neural Networks, Science China Information Sciences, and the Frontiers of Computer Science. He is the Steering Committee member of the ACML and PAKDD, the vice chair of the CAAI Machine Learning Society, and the Standing Committee member of the CCFArtificial Intelligence and Pattern Recognition Society. He is also a distinguished member of CCF, CAAI, and a senior member of ACM.

Cheng-Yu Liu (Senior member, IEEE) received Ph.D. degrees in Biomedical Engineering from Shandong University, China, in 2010. He is now a Professor of the State Key Laboratory of Bioelectronics, and the founding Director of Wearable Intelligent Monitoring Lab in Southeast University. He has published more than 300 original Journal/Conference papers, and holds more than 30 patents as an inventor. His research topics include: wearable medicine and intelligent monitoring.