Self-Training with Decoupled Label Correction for Noisy Label Learning

Anonymous CVPR submission

Paper ID 8022

Abstract

The high cost of acquiring large-scale, accurately labeled datasets has made the efficient use of noisy data crucial for supervised learning. Studies have shown that incorporating a small amount of clean data with various strategies can improve learning outcomes. In this study, we introduce a novel methodology for leveraging limited quantities of clean data to assist learning from noisy labels. Our method departs significantly from traditional approaches, which often rely on estimating a transition matrix or meta-gradient adjustments. Instead, our approach directly utilizes limited clean data to guide the correction of noisy labels through decoupled bi-level optimizations in a self-training framework, where the classifier and label correction function iteratively reinforce each other. Our method achieves advanced performance on the Clothing1M dataset without fine-tuning on clean data, with lower overheads than leading methods, demonstrating its effectiveness in real-world applications.

1. Introduction

The success of deep neural networks in various domains highlights the importance of large-scale data sources in achieving high-performing models. However, obtaining a completely clean dataset for real-world production can often be prohibitively expensive. When direct supervised learning is applied to noisy datasets, the neural network tends to overfit the noise present in the data. This phenomenon leads to a degradation in model performance and generalization [29, 42]. Therefore, learning with label noise (LNL) has become a critical area of research, and multiple methods have been proposed to address this issue.

While it is challenging to obtain an entirely clean dataset, it is still feasible to accurately label a small part of the data. Some research has shown significant performance improvements when a small amount of reliable data is used to assist training [16]. With a small amount of clean dataset, meta-learning methods have proven effective in improving final performance over time. They usually optimize hyperparameters on clean data through nested bi-level optimization. However, these methods also have notable limitations. Their dependency on clean data often involves implicit optimization, which can restrict model interpretability and lead to an over-reliance on previously miscorrected labels [13]. Additionally, a significant drawback is the requirement for multiple virtual updates within the inner loop, resulting in substantial computational overhead [32, 43] and increased memory usage, which limits their efficiency and feasibility for large-scale datasets. Transition matrix-based methods are also widely used to leverage clean data. They model the label corruption process with a matrix mapping from the clean distribution to the noisy distribution, and try to use the clean data to estimate it. However, modeling instance-dependent corruption remains challenging, though some works have attempted to address this issue [40, 47]. Furthermore, despite having classifier-consistent properties [36], these methods still struggle with high noise variance. Individual samples can deviate significantly from the expected corruption pattern, which can be detrimental to the optimization process.

If feasible, directly recovering noisy labels into the clean ones can be effective. However, if using the transition matrix method, it involves computing the matrix inverse, which can be computationally intensive and may lead to instabilities. Some meta-learning-based methods attempt to correct noisy labels through a meta-process but suffer from high computational costs [30, 46]. The DMLP method [31] tries to recover labels by performing a linear transformation on contrastive learning features. However, this approach requires additional self-supervised learning steps, resulting in further time costs, and the assumption of linear transformation on the self-supervised-learned features may not always be suitable.

To address the aforementioned challenges, we propose an innovative label correction technique based on a decoupled bi-level optimization framework in a self-training style. This framework enables the correction function to co-evolve with the classifier, allowing it to refine noisy labels and thereby enhance the classifier's accuracy. Our approach introduces a new paradigm for utilizing a small amount of clean data in noisy label learning, with a label correction function learning method that differs from traditional meta-

learning approaches. While our method shares similarities with transition matrix approaches in directly learning the relationship between noisy and clean labels, it does not suffer from the constraints of transition matrix approaches.

Our algorithm leverages the network's intrinsic tendency to fit clean labels, providing key insights for label correction. This characteristic is also fundamental to pseudo-labeling techniques, which generate pseudo-labels from network predictions [4, 15]. Building on this intuition, we make further explorations. During forward propagation, information is gradually compressed, reaching its minimal state after the final activation. Therefore, we use mid-layer features rather than final layer outputs to increase information, enabling more precise label correction.

We validate our approach with comprehensive experiments on CIFARs and Clothing1M datasets. The results demonstrate that our method achieves notable performance improvements and significantly reduces training time compared to previous methods. This advancement provides a promising and efficient approach for large-scale noisy label learning, effectively utilizing limited clean data to refine noisy labels in real-world scenarios.

2. Related Work

2.1. Various Methods for Noisy Label Learning

A broad range of studies have explored strategies to mitigate the impact of noisy labels, employing diverse techniques and perspectives. Theoretically analyzing the impact of label noise has provided insights into when and why noise hurts performance [1, 2, 22]. Many works have developed noise-robust loss functions and regularization to prevent overfitting [18, 33, 44]. Reweighting techniques [10, 19] aim to reduce the influence of likely mislabeled examples. Sample selection methods identify high-confidence clean labels to enhance training robustness [35, 37]. Some studies have focused on enhancing optimization techniques to alleviate the impact of noise [7, 17]. Semi-supervised learning leverages unlabeled or noisy data to combat label noise [8, 25, 26]. More recent efforts [5, 6, 20] tackle instance-dependent label noise, which better matches real-world settings.

2.2. Noisy Label Learning with Clean Data Subset

Recent research has shifted focus toward two main directions: learning noise transition matrices and utilizing meta-learning techniques to optimize performance on clean validation data.

The transition matrix approach, powered by forward loss correction [23] re-weighting mechanisms [36], offers classifier-consistency guarantees when provided with sufficient samples and accurate transition matrices. These methods leverage clean data to estimate the transition patterns from noisy to clean labels [9, 13], and have been successfully extended to multi-label classification scenarios [24].

In the meta-learning paradigm, several approaches have emerged. Some methods [19, 27, 43] employ sample re-weighting strategies to mitigate the impact of noisy labels on the training loss. Others utilize meta-optimization to estimate noise transition matrices [28, 34]. Some approaches [11, 45] meta-learn optimal weights for pseudo-labels in self-training scenarios. Additionally, the student-teacher framework has been adopted by various methods [14, 30, 34], albeit with different implementation details. Famus [39] makes a good attempt to accelerate the meta-learning process and also achieve a low-variance approximation of meta-gradient.

3. Preliminaries

In this study, we address the problem of c-class classification. The input feature space is represented as $x \in \mathbb{R}^d$, and the label space is denoted as $y \in \mathbb{R}^c$, where c signifies the number of classes.

We define the noisy training dataset as $\tilde{\mathcal{D}} = \{(x_i, \tilde{y}_i)\}_{i=1}^N$, where each $\tilde{y}_i$ is an observed noisy label. This dataset is assumed to be sampled from a noisy label distribution $p_{x,\tilde{y}}$. Correspondingly, the latent true dataset for $\tilde{\mathcal{D}}$ is denoted by $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, containing the true labels for each feature vector x_i. Each sample in $\tilde{\mathcal{D}}$, $(x_i, \tilde{y}_i)$, is paired with a possibly noisy label $\tilde{y}_i$, while in $\mathcal{D}$, the same feature x_i is associated with its true label y_i. We also incorporate a subset of trusted data, referred to as the clean validation dataset $\mathcal{D}^{\text{m}} = \{(x_i^m, y_i^m)\}_{i=1}^M$, sourced from the clean label distribution $p_{x,y}$. Usually, $M \ll N$ holds, as is common in real-world scenarios.

Let $f(x;\theta)$ be a neural network model parameterized by θ for classification. We assume that the model can be decomposed as $f(x;\theta) = g(e(x;\theta^e);\theta^c)$, where $\theta = \{\theta^e, \theta^c\}$. Specifically, the feature extractor $e(x;\theta^e)$ maps the input x to a feature representation $z \in \mathbb{R}^{d'}$, where d' is the dimension of the feature embeddings, with θ^e denoting the parameters of the extractor. We assume that $d' \ll d$ where d is the dimension of x. The prediction head parameterized with θ^c then outputs class probabilities via a softmax layer, approximating the class-conditional distribution $p(y|x)$. Our objective is to learn a classifier that performs well on clean data, even when trained primarily on noisy labels.

4. Methods

4.1. Decoupled Label Correction Formulation

To address the LNL problem, the goal of our framework is to develop a correction function h parameterized by ψ that can transform noisy labels $\tilde{y}$ into the underlying clean labels y. We consider instance-dependent scenario for the general case, where noisy label is corrupted from the label y, and may also be influenced by the data x. We define the approximated clean labels at each step as $\hat{y} = h(x, \tilde{y}; \psi)$

and aim for these labels to approach the latent true labels. The corrected labels are used to train a classifier to improve performance on clean data, described as

$$\theta^* = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}_{\text{train}}(f(x_i;\theta), h(x, \tilde{y}; \psi)). \quad (1)$$

To learn this correction function, we leverage the clean dataset $\mathcal{D}^{\text{m}}$ to optimize the function h. This optimization is achieved by minimizing the loss function on the clean dataset with respect to the correction function parameters ψ:

$$\psi^* = \arg\min_{\psi} \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_{\text{meta}}(h(x^m, \hat{\tilde{y}}^m; \psi), y_i^m), \quad (2)$$

where the definition of $\hat{\tilde{y}}^m$ is the approximated noisy posterior as detailed below.

4.2. Noisy Posterior on the Clean Dataset

Particularly, the correction function h takes $\tilde{y}$ as input, using the information of how y are corrupted into $\tilde{y}$ to help correct them. However, in the clean dataset where h is learned, noisy labels are not available. Therefore, we simulate a noisy posterior, $\hat{\tilde{y}}^m$, as a proxy for noisy labels on this clean data to enable h_t to function consistently across both noisy and clean datasets.

To model noisy posteriors, we introduce an additional prediction head after the feature extractor, which is parameterized by θ^n. We configure the noisy prediction head the same way as the clean prediction head parameterized with θ^c. For the rest of the paper, the modified network parameters are represented as $\theta = \{\theta^e, \theta^c, \theta^n\}$. This noisy prediction model is denoted as $f_n(x;\theta) = g(e(x;\theta^e);\theta^n)$, and it is trained directly on noisy label instances $\{(x_i, \tilde{y}_i)\}_{i=1}^N$, to capture noise patterns in the data.

To generate noisy posterior predictions for the clean dataset, for each sample x_i^m in $\mathcal{D}^{\text{m}}$, we compute $\hat{\tilde{y}}_i^m = f_n(x_i^m)$, which provides the model's predicted probability distribution to approximate how noise might appear on the clean dataset. This strategy enables the function h_t to operate consistently across both noisy and clean datasets, leveraging simulated noisy labels $\hat{\tilde{y}}^m$ on the clean data.

4.3. Neural Label Correction

Previous research has demonstrated that neural networks inherently prioritize learning patterns from clean labels. This observation has led to various self-learning approaches that leverage pseudo-labels or soft labels generated by the classifier itself [3, 21, 34]. However, these pseudo-labels may contain inaccurate information. Moreover, due to the information shrinkage phenomenon, features from earlier network layers retain richer semantic content compared to later layers.

Algorithm 1 Learning ψ^* of Neural Label Correction

Require: Clean training dataset $\mathcal{D}^{\text{m}}_{\text{train}}$; clean validation dataset $\mathcal{D}^{\text{m}}_{\text{val}}$, noisy posterior $\hat{\tilde{y}}_i^m$, and the feature set $\mathcal{Z}^m = \{z_k^m\}_{k=0}^K$.
Ensure: Optimized ω^*, α^*

1: Initialize the neural network $r(\cdot,\omega)$ and the ratio α^*.
2: Train $r(\cdot;\omega)$ on $\mathcal{D}^{\text{m}}_{\text{train}}$ by minimizing:

$$\frac{1}{|\mathcal{D}^{\text{m}}_{\text{train}}|} \sum_{i \in \mathcal{D}^{\text{m}}_{\text{train}}} \mathcal{L}_{\text{meta}} \left(r\left(\hat{\tilde{y}}_i^m, \{z_{k,i}^m\}_{k=0}^K; \omega\right), y_i^m \right).$$

3: Select ω^* on $\mathcal{D}^{\text{m}}_{\text{val}}$ by minimizing:

$$\frac{1}{|\mathcal{D}^{\text{m}}_{\text{val}}|} \sum_{i \in \mathcal{D}^{\text{m}}_{\text{val}}} \mathcal{L}_{\text{meta}} \left(r\left(\hat{\tilde{y}}_i^m, \{z_{k,i}^m\}_{k=0}^K; \omega\right), y_i^m \right).$$

4: Optimize α^* on $\mathcal{D}^{\text{m}}_{\text{val}}$ by minimizing:

$$\frac{1}{|\mathcal{D}^{\text{m}}_{\text{val}}|} \sum_{i \in \mathcal{D}^{\text{m}}_{\text{val}}} \mathcal{L}_{\text{meta}} \left(\alpha \hat{\tilde{y}}_i^m + (1-\alpha) \bar{y}_i^m, y_i^m \right),$$

where

$$\bar{y}_i^m = r\left(\hat{\tilde{y}}_i^m, \{z_{k,i}^m\}_{k=0}^K; \omega^*\right).$$

return ω^*, α^*

This characteristic suggests that leveraging features from earlier layers could enhance the effectiveness of label correction mechanisms.

In our method, we also utilize the previously learned classifier $\{\theta_1, \theta_2, \ldots, \theta_K\}$ to guide correction, where each θ_k represents the neural network parameters at iteration k. In practical settings, the number of clean instances available in the clean dataset $\mathcal{D}^{\text{m}}$ is often limited. This limitation poses challenges for optimizing the parameters at the scale of historical classifier parameters $\{\theta_k\}$ effectively. As a result, we only use the frozen feature extractor components $\{\theta_k^e\}$ from previous iterations, which capture more identifiable information about the input data within the extracted features.

We introduce a simple and shallow neural network parameterized by ω to recover the corrected labels. The recovered label, denoted by $\bar{y}$, is produced by the prediction function:

$$\bar{y} = r\left(\tilde{y}, \{z_k\}_{k=0}^K; \omega\right), \quad (3)$$

where $z_k = e(x;\theta_k^e)$. Here, r represent the neural network function that takes the noisy label $\tilde{y}$ and the accumulated feature representations as inputs to predict the estimated label after the softmax function. The neural network input dimension is $c + (K+1) \cdot d'$ and the output dimension is fixed at c as the label dimension. Due to the limited size of the clean dataset, directly training the correction function

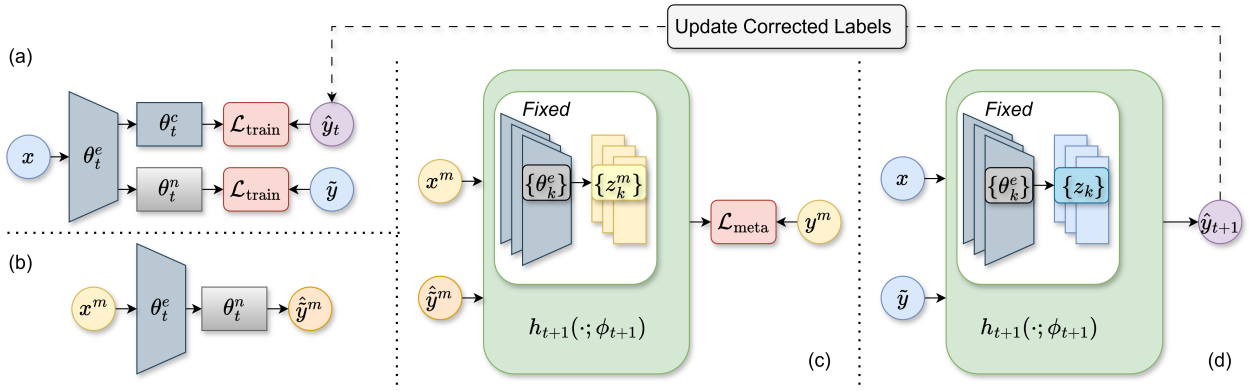

Figure 1. This framework provides a general overview of the approach. (a) Training the classifier on the corrected labels. (b) Simulate the noisy posterior on the clean dataset. (c) Optimization of the correction function. (d) Noisy label correction.

on $\mathcal{D}^{\text{m}}$ can lead to overfitting. To mitigate this, we split $\mathcal{D}^{\text{m}}$ into a correction training set $\mathcal{D}^{\text{m}}_{\text{train}}$ and a validation set $\mathcal{D}^{\text{m}}_{\text{val}}$. The correction function is trained on $\mathcal{D}^{\text{m}}_{\text{train}}$ and validated on $\mathcal{D}^{\text{m}}_{\text{val}}$, selecting the model parameters that achieve the best performance on the validation set to ensure generalization.

Furthermore, we introduce a mixing ratio vector α, optimized on the validation set, to combine the original noisy labels with the predicted labels from the correction function in a convex manner:

$$\hat{y} = \alpha\tilde{y} + (1-\alpha)\bar{y}, \tag{4}$$

where $\hat{y}$ represents the final corrected labels. This convex combination ensures that even if the predicted labels have the worst generalization performance on other clean data, the corrected labels $\hat{y}$ are still expected to be cleaner than the original noisy labels.

The parameters ψ in our neural label correction method consist of two components: ω for the correction network and α for the mixing coefficient. The correction function h combines the original noisy label with a corrected version predicted with the neural network. The detailed algorithm for optimizing the correction parameters is presented in Algorithm 1. By leveraging the accumulated feature representations and using a simple neural network as the correction function, this practical algorithm effectively refines label quality in the training data. The convex combination with the mixing ratio α ensures robustness against potential inaccuracies in the correction function's predictions, guaranteeing that the corrected labels quality.

4.4. Self-Training Label Correction Framework

In this section, we introduce our iterative label correction framework that jointly optimizing the classifier on the corrected label and the label correction function. A general overview of our method is provided in Fig. 1.

In our algorithm, we initialize the target labels with the noisy labels, although alternative initialization strategies such as label smoothing or other correction techniques can be applied based on prior knowledge. After that, the iterative process comprises the following key components:

Classifier Update: At each iteration t, we update the classifier parameters θ_t by minimizing the training loss $\mathcal{L}_{\text{train}}$ over the corrected labels $\hat{y}_{t,i}$ with

$$\theta_t = \arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}\Big(\mathcal{L}_{\text{train}}(f(\mathbf{x}_i;\theta),\hat{y}_{t,i}) + \lambda\mathcal{L}_{\text{train}}(f_n(\mathbf{x}_i;\theta),\tilde{y}_i)\Big), \tag{5}$$

where λ is a loss balancing factor to prevent the model from overfitting to noise. This step aims to improve the classifier's ability to predict the corrected labels accurately.

Update the Simulated Noisy Posterior Proxy: To enable the correction function to utilize noisy label information on the clean dataset, we update the simulated noisy posterior $\hat{\tilde{y}}^m$ by applying the noisy prediction model f_n to the clean data x^m.

Feature Collection: To avoid redundant computations for the same historical classifier in different iterations, we extract features set $\mathcal{Z} = \{z_k = e(x;\theta^e_k)\}^t_{k=0}$ and $\mathcal{Z}^m = \{z^m_k = e(x^m;\theta^e_k)\}^t_{k=0}$ by the extractors up to iteration t so that the elements can be reused,.

Correction Function Optimization: With the simulated noisy posteriors and the accumulated feature sets, we initialize and optimize the correction function parameters $\psi_{t+1} = \{\omega_{t+1}, \alpha_{t+1}\}$ by minimizing the meta loss $\mathcal{L}_{\text{meta}}$ over the clean data distribution with Algorithm 1. This step refines the correction function to better approximate the true labels.

Noisy Label Correction: The updated correction function h_{t+1} parameterized ψ_{t+1} is then applied to compute new corrected labels $\hat{y}_{t+1,i}$, which will be used for classifier

training in the next iteration, described by

$$\hat{y}_{t+1,i} \leftarrow \alpha_{t+1}\tilde{y}_i + (1-\alpha_{t+1}) r_{t+1}\left(\tilde{y}_i, \{z_{k,i}\}_{k=0}^{t}; \omega_{t+1}\right), \quad i = 1, \ldots, N. \tag{6}$$

The complete bi-level self-training procedure is outlined in Algorithm 2. By iteratively updating both the classifier and the correction function, our framework effectively leverages the interplay between the noisy training data and the clean data. This iterative process continues until the optimized classifier is robust to label noise.

4.5. Advantages of the Proposed Method

The proposed technique provides several advantages over previous methods which typically utilize clean data in more conventional ways. These advantages include:

- **Optimization Efficiency:** Traditional meta-gradient approaches often involve high computational costs due to the stepwise updates. Especially in multi-GPU training scenarios, computing high-order derivatives is challenging. Our decoupled objectives, however, allows for update of h_t after multiple epochs update for the classifier. It also facilitates seamless scaling to distributed setups, ultimately enhancing the overall training efficiency.
- **Flexible Adjustments:** In contrast to traditional meta-learning methods where the forms of meta-parameters are typically fixed, our approach allows for dynamic adjustment of parameters within the function h. This flexibility provides a more adaptable framework, capable of handling various data conditions throughout the training process.
- **Adaptability and Robustness:** Our approach easily handle instance-independent noise and may also adapt to elegantly handle continuous or multi-label problems which would be hard to modeled with transitions matrix, showcasing its versatility.

Algorithm 2 Self-Training with Neural Label Correction

Require: Noisy training dataset $\tilde{\mathcal{D}} = \{(x_i, \tilde{y}_i)\}_{i=1}^{N}$; clean dataset $\mathcal{D}^{\text{m}} = \{(x_i^m, y_i^m)\}_{i=1}^{M}$.

Set iteration counter $t \leftarrow 0$.
Initialize target labels $\hat{y}_{0,i} \leftarrow \tilde{y}_i$.
Initialize sets of extracted features: $\mathcal{Z} \leftarrow \emptyset$, $\mathcal{Z}^m \leftarrow \emptyset$.
Split $\mathcal{D}^{\text{m}}$ into $\mathcal{D}^{\text{m}}_{\text{train}}$ and $\mathcal{D}^{\text{m}}_{\text{val}}$.
repeat
  Update classifier parameters θ_t with Eq. (5).
  $\hat{\tilde{y}}_i^m \leftarrow f_n(x_i^m; \theta_t), \quad z_{t,i}^m \leftarrow e(x_i^m; \theta_t^e), \quad i = 1, \ldots, M.$
  $z_{t,i} \leftarrow e(x_i; \theta_t^e), i = 1, \ldots, N.$
  $\mathcal{Z} \leftarrow \mathcal{Z} \cup \{z_{t,i}\}_{i=1}^{N}, \mathcal{Z}^m \leftarrow \mathcal{Z}^m \cup \{z_{t,i}^m\}_{i=1}^{M}$.
  Update $\omega_{t+1}, \alpha_{t+1}$ with Algorithm 1.
  Update $\hat{y}_{t+1,i}, i = 1, \ldots, N$ with Eq. (6).
  $t \leftarrow t + 1$.
until Done

5. Experiments

5.1. Experiments on CIFARs

We first evaluate our self-learning neural label correction method on CIFAR-10 and CIFAR-100 [12] which is widely used for machine learning research. Both datasets contain 50,000 training and 10,000 test images. We randomly select 5,000 images from the training set as the clean dataset.

Noise settings: Let η denote the corruption ratio. We simulate two types of **Instance-independent** noise. **Symmetric**: Generated by flipping the labels of a given proportion of training samples to all the other class labels uniformly. Each instance with a true label y is corrupted to all possible classes with probability $\frac{\eta}{c}$. **Asymmetric**: Generated by flipping the labels of a given proportion to one or few particular classes. For CIFAR-10, we disturb the label to its similar class with probability η follow [41]. For CIFAR-100, we disturb the label to other classes in its super-class as described in[9]. Noise transition matrices are first generated and labels are randomly corrupted according to the matrix for instance-independent label noise. Additionally, we evaluate **Instance-dependent** noise as described in [47] on both datasets. We vary η in the range of $[0, 1]$ to simulate different noise levels for all noise types.

Baselines: We compare our methods with a series of advanced methods in noisy label learning that use clean data. The compared methods include: *Transition-Matrix-Based Methods*: **GLC.** [9] estimates the noise transition matrix with a trained noisy posterior. **FasTEN.** [13] introduce a two-head architecture to efficiently estimate the label transition matrix every iteration within a single back-propagation. *Meta-Learning-Based Methods*: **MW-net.** [27] uses an MLP net and meta-learns the weighting function. **L2B.** [11] uses a bootstrapping method with meta-learned weighting coefficient. **EMLC.** [30] proposes a teacher-and-student architecture and meta-learns the taught soft-label. Notably, for FasTEN and GLC, we modified their standard training procedure by removing the loss component corresponding to the clean instances, in order to maintain experiment consistency. For EMLC, we remove the self-supervised learning initialization for a fair comparison.

Implementations: In our experiments on CIFAR-10 and CIFAR-100, we employed ResNet-34 as the backbone architecture. To ensure a fair comparison across different methods, each was trained from scratch under identical conditions. For both datasets, we applied horizontal flip and random crop as data augmentations. The models were trained using stochastic gradient descent (SGD) optimizer with momentum 0.9 and weight decay $5e^{-4}$ for 100 epochs, and the batch size

CVPR #8022

Table 1. Test accuracy (%) of all competing methods on CIFAR-10 and CIFAR-100 under different noise types with different noise levels. The best results are highlighted in bold.

Dataset	Method	Symmetric			Asymmetric			Instance-dependent		
		η			η			η		
		0.2	0.4	0.6	0.2	0.4	0.6	0.2	0.4	0.6
CIFAR-10	Cross-entropy	83.88	72.09	57.17	88.76	81.7	69.05	77.89	56.32	39.80
	GLC	88.59	83.44	76.43	91.37	89.78	86.72	88.57	84.35	63.42
	FasTEN	86.19	77.62	70.04	91.63	88.82	85.71	86.82	81.16	61.88
	MW-Net	84.86	72.58	57.04	90.49	87.21	61.65	79.29	66.94	52.41
	L2B	88.83	86.27	80.23	91.23	90.51	89.23	89.46	85.79	74.23
	EMLC	81.60	71.24	58.55	88.51	80.95	68.11	80.32	74.09	61.57
	Ours	**90.53**	**87.42**	**80.46**	**92.93**	**91.77**	**91.10**	**91.19**	**88.23**	**80.98**
CIFAR-100	Cross-entropy	62.34	46.40	28.16	63.04	48.05	29.94	58.75	40.57	22.00
	GLC	65.75	52.03	41.16	67.57	58.84	37.64	60.63	55.53	45.29
	FasTEN	59.37	43.65	26.09	61.91	43.27	31.50	59.70	45.79	33.15
	MW-Net	62.38	48.59	29.55	62.87	47.82	31.95	64.89	40.91	22.75
	L2B	64.28	54.38	20.66	64.66	58.11	43.33	66.05	59.55	47.45
	EMLC	62.80	46.51	31.74	62.94	47.09	32.34	61.94	49.34	28.44
	Ours	**66.67**	**61.79**	**52.06**	**69.68**	**65.39**	**62.44**	**67.71**	**63.22**	**53.68**

was set to 128. We set the initial learning rate to 0.1 with a step decay scheduler that reduces the learning rate by a factor of 0.1 at epochs 60 and 80.

For our method's specific settings, we make the loss balancing factor $\lambda = 0.5$ in Eq. (5). We split the dataset $\mathcal{D}^{\text{m}}$ into training $\mathcal{D}^{\text{m}}_{\text{train}}$ and validation subsets $\mathcal{D}^{\text{m}}_{\text{val}}$ using an $0.8 : 0.2$ ratio. The neural network correction model is a 2-layer multilayer perceptron (MLP) with 256 hidden units, ReLU activation, and a softmax output layer. We make $\mathcal{L}_{\text{meta}}$ to be Cross-Entropy and use SGD with learning rate 0.001 on $\mathcal{D}^{\text{m}}_{\text{train}}$ to optimize the ω, while we employ sequential least squares programming to optimize α on $\mathcal{D}^{\text{m}}_{\text{val}}$. We update the feature sets and correction function every 8 epochs, starting after the initial 40 epochs, to allow for an adequately initialized noisy posterior.

To accurately assess the models' ability to handle label noise and avoid overfitting to it, we report the final accuracies for each experiment without the use of early stopping or other similar techniques.

5.1.1. Results

The overall results are displayed in Tab. 1. Generally, our method consistently outperforms other approaches across all noise types, demonstrating strong robustness to label noise. Notably, our method significantly outperforms transition matrix-based methods even under instance-independent noise. This suggests that our neural network correction function effectively reduces the variance within the noisy labels, which can otherwise undermine the performance of transition matrix-based methods. We demonstrate the training dynamics comparison for CIFAR-100 under instance-dependent noise with a noise level of 0.4 in Fig. 2. After the first label correction at epoch 40, there is a notable increase in accuracy. During training, MW-Net and FasTEN exhibit declines in performance in the later stages, likely due to overfitting to noise patterns. In contrast, our method shows no significant signs of overfitting, indicating that it effectively leverages limited clean data to improve training on noisy labels under various noise conditions.

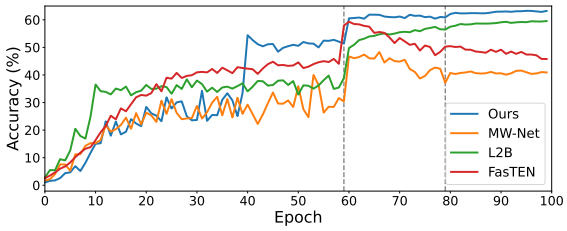

Figure 2. Training dynamics of different methods on CIFAR-100 under instance-dependent noise with a level of 0.4. The vertical dashed lines represent the epochs at which the learning rate decays.

5.1.2. Training Time Consumption Comparison

For our experiments on CIFARs, all models were trained using a single RTX 4090 GPU. We have quantified the training time for our method and compared it with the baseline methods. These results are presented in Tab. 2. Our approach secures a leading position in terms of training efficiency, necessitating less time compared to existing meta-learning-based methods.

5.1.3. Ablations for the Correction Module

In this experiment, we test three hypotheses. The first hypothesis concerns information shrinkage, suggesting that features extracted from earlier layers could contribute more meaningfully to label correction. To test this, we use the

Table 2. This table details the average GPU training time on CIFAR-100 (in hours) for various methods using a single RTX 4090, with the relative ratio of time compared to our method.

METHOD	MW-NET	L2B	EMLC	GLC	FASTEN	OURS
TOTAL TRAINING TIME (RELATIVE TO OURS)	1.08 (3.38X)	1.61 (5.03X)	1.09 (3.41X)	0.63 (1.97X)	**0.31 (0.97X)**	**0.32**

Table 3. Test accuracy (%) of ablation studies on CIFAR-100. A1: Use last-layer logit to replace the features; A2: Linear transformation on the features; A3: Use only latest features for correction. The best results are in bold.

METHOD	SYMMETRIC			ASYMMETRIC			INST-DEPENDENT		
	η			η			η		
	0.2	0.4	0.6	0.2	0.4	0.6	0.2	0.4	0.6
A1	65.72	58.38	47.78	67.64	61.83	55.33	66.39	58.36	47.50
A2	65.86	58.94	48.81	68.09	62.89	57.25	66.68	60.22	48.84
A3	65.09	57.99	47.95	66.73	63.30	58.52	**69.32**	62.03	**55.61**
OURS	**66.67**	**61.79**	**52.06**	**69.68**	**65.39**	**62.44**	67.71	**63.22**	53.68

network's final logits (before the softmax operation) instead of earlier-layer features. As a result, the input dimension of r_t becomes $(1+t)\cdot c$ at each iteration $t \geq 1$. The second hypothesis examines whether a linear transformation followed by a softmax activation, similar to the approach in DMLP, is appropriate compared to our two-layer MLP with non-linear transformations. For this, we define the corrected noisy label as $\hat{y}_t = \alpha\tilde{y} + (1-\alpha)\sigma_{sm}(T_t\left[z_0, \ldots, z_{t-1}\right])$ for $t \geq 1$, where T_t is a trained linear mapping from d' to c and σ_{sm} is the softmax function. The third hypothesis explores whether using all previous representations enhances performance compared to using only the latest one, even though the most recent representation theoretically captures the most separable features. To test this, we input only $\tilde{y}$ and z_t into the neural network at iteration t.

We evaluate these three hypotheses using the CIFAR-100 dataset, with results presented in Tab. 3. The findings indicate that the first hypothesis is supported, suggesting that pseudo-labeling could benefit from relying on final outputs, even though information may be diminished after the softmax function. The ablation results further validate our second hypothesis. The linear transformation lacks the expressive capacity needed to capture complex label relationships. Non-linear transformations provide a distinct advantage in label correction by enabling a richer representation of the noisy-to-clean label mapping. The third ablation results reveal that using features from only the latest layer results in slightly lower accuracy in more cases. This result suggests that cumulative feature information can enhance label correction by incorporating a broader range of representations.

5.1.4. Ablations for the Loss Balancing Factor

In addition, we also conduct experiments to examine how the loss balancing factor λ affects performance. Our method requires an accurate estimation of the noisy posterior on the clean dataset. Selecting the appropriate λ in Eq. (5) is a balancing act: it should enable the classifier to estimate the noisy posterior accurately, while preventing the feature extractor from overfitting to noisy patterns. We vary λ across different values in $\{0.2, 0.5, 1.0\}$, with the results summarized in Tab. 4.

Table 4. Test accuracy (%) for different λ values on CIFAR-100. The best results will be highlighted in bold.

λ	SYMMETRIC			ASYMMETRIC			INST-DEPENDENT		
	η			η			η		
	0.2	0.4	0.6	0.2	0.4	0.6	0.2	0.4	0.6
0.2	**68.13**	61.74	50.30	**69.88**	**67.62**	**63.86**	**69.13**	62.69	51.14
0.5	66.67	**61.79**	**52.06**	69.68	65.39	62.44	67.71	**63.22**	**53.68**
1.0	67.25	59.98	50.49	69.65	66.25	61.44	68.48	61.07	51.22

5.2. Experiments on Clothing1M

Clothing1M [38] is a real-world large-scale dataset where image instances come from online shopping websites with noisy labels derived from user tags. For these experiments, we used ResNet-50 pre-trained on ImageNet as our backbone, following previous works. The model was trained for 15 epochs using SGD optimizer with momentum 0.9, weight decay $5e^{-4}$, and batch size 256. We set the initial learning rate to 0.01 with step decay at epochs 5 and 10. The training pipeline included more data augmentations: resized crop, horizontal flip, random color jitter, and random grayscale conversion following previous works.

All label correction parameters remain consistent with previous experiments, maintaining the loss balancing factor $\lambda = 0.5$. However, we adjusted the correction function update frequency to every epoch due to the dataset's scale. In addition to the baseline methods, we also consider other leading approaches [31, 46] of label correction for comparison.

5.2.1. Results

We display our training dynamic in Fig. 3 and the comparison result is presented in Tab. 5. The corrected labels produced by h are more accurate at the beginning, while the classifier accuracy gradually catches up as training progresses, using the generated labels. Our approach achieves the highest accuracy of 79.82%, outperforming other leading methods, including EMLC (79.35%) and DMLP-DivideMix (78.23%), even though these methods use additional self-supervised

Table 5. Comparative of test accuracy on Clothing1M with advanced methods utilizing clean data. The symbol ✦ indicates the use of meta-knowledge, ✓ denotes methods that directly train on clean instances with full forward and back propagations on the classifier parameters including θ^e and θ^c, and ♠ denotes methods that require additional self-supervised learning procedures.

METHOD	EXTRA DATA	ACCURACY (%)
CROSS-ENTROPY	✗	69.21
GLC	✓	74.26
FASTEN	✓	77.83
MW-NET	✦	73.72
MLC	✦	75.78
DMLP-DIVIDEMIX♠	✦	78.23
EMLC♠	✦	79.35
OURS	✦	**79.82**

Table 6. Test accuracy (%) of our method and DMLP under varying different sizes of clean dataset. The results of DMLP is borrowed from their paper [31]. The symbol ♠ denotes method that requires additional self-supervised learning procedures.

METHOD	SIZE OF CLEAN DATASET		
	10%	50%	100%
DMLP♠	75.50	77.30	77.31
OURS	78.70	79.76	**79.82**

learning techniques. It indicates that our method provides a more reliable approach for learning with noisy labels in real-world, large-scale scenarios.

5.2.2. Training Efficiency Comparison

In order to validate our potential in practical usage, we test the training efficiency for the advanced methods, mainly the meta-learning-based methods, on Clothing1M. We set up comparative analysis of our method, L2B and EMLC regarding their time cost and performance. We unify the training configuration to ensure a fair basis for comparison. Our method and L2B utilize a single RTX 4090 GPU with 128 mini-batches, whereas EMLC employs a 2-GPU setup, with each GPU processing 64 mini-batches. We unify the classifier backbone as Resnet-50, and remove the self-supervised learning session for EMLC. The other configurations for our method remain the same with our settings and their original reports. This configuration ensures a fair basis for comparison.

The training time comparison is presented in Fig. 4. Notably, the other two meta-learning-based methods required significantly more time per epoch, while our framework demonstrated superior efficiency. Additionally, our method used less memory, with EMLC consuming more than twice the memory of ours. These results highlight the advantages

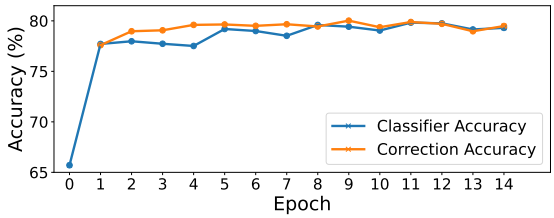

Figure 3. The training dynamic of our method on Clothing1M. The classifier accuracy is evaluated on the test dataset, while the correction function accuracy is evaluated on $\mathcal{D}_{\text{val}}^{\text{m}}$.

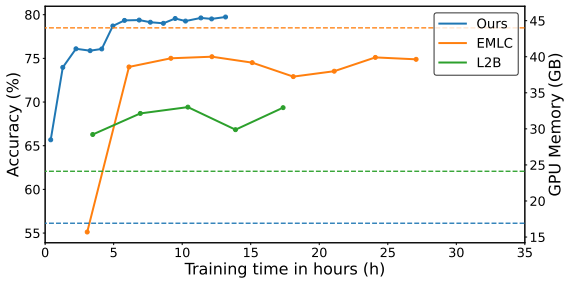

Figure 4. Comparison of training time, memory usage (GB), and test accuracy (%) on Clothing1M. The plot shows the training process for our method, L2B, and EMLC. The solid line represents test accuracy, while the dashed line indicates GPU memory usage during training.

of our approach in terms of performance, time efficiency, and GPU memory usage, underscoring its potential for practical applications.

5.2.3. Impact of the size of clean data

We explored the impact of the clean data scale on our performance. This aspect was examined using two different sizes of clean sample sets: 10% and 50% samples. The corresponding results are detailed in Tab. 6. Our method consistently performs well across all clean data sizes. With only 10% of the clean data, our approach still achieves an accuracy of 78.70%, surpassing DMLP by a margin of 3.20%. These findings demonstrate that our method is capable of effectively utilizing limited clean data, achieving competitive performance even with smaller subsets.

6. Conclusion

This study presents a novel approach to correcting noisy labels in noisy label learning, with a focus on effectively utilizing limited clean data. At the core of our contribution is the development of a self-training framework that incorporates decoupled bi-level optimization, guiding the label correction process using clean data. Our experiments on different benchmarks validate the effectiveness and efficiency of our method. This approach not only improves accuracy in noisy label scenarios but also reduces computational overhead, offering a robust solution for practical applications.

References

[1] Julius Adebayo, Melissa Hall, Bowen Yu, and Bobbie Chern. Quantifying and mitigating the impact of label errors on model disparity metrics. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[2] Oshrat Bar, Amnon Drory, and Raja Giryes. A spectral perspective of dnn robustness to label noise. In *International Conference on Artificial Intelligence and Statistics*, pages 3732–3752. PMLR, 2022. 2

[3] WANG Botao, Jia Li, Yang Liu, Jiashun Cheng, Yu Rong, Wenjia Wang, and Fugee Tsung. Deep insights into noisy pseudo labeling on graph data. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 3

[4] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv preprint arXiv:2301.10921*, 2023. 2

[5] Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. Beyond class-conditional assumption: A primary attempt to combat instance-dependent label noise. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11442–11450, 2021. 2

[6] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. *arXiv preprint arXiv:2010.02347*, 2020. 2

[7] Eduard Gorbunov, Marina Danilova, and Alexander Gasnikov. Stochastic optimization with heavy-tailed noise via accelerated gradient clipping. *Advances in Neural Information Processing Systems*, 33:15042–15053, 2020. 2

[8] Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5138–5147, 2019. 2

[9] Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. *Advances in neural information processing systems*, 31, 2018. 2, 5

[10] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pages 2304–2313. PMLR, 2018. 2

[11] Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9676–9686, 2022. 2, 5

[12] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[13] Seong Min Kye, Kwanghee Choi, Joonyoung Yi, and Buru Chang. Learning with noisy labels by efficient transition matrix estimation to combat label miscorrection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV*, pages 717–738. Springer, 2022. 1, 2, 5

[14] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Learning to learn from noisy labeled data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5051–5059, 2019. 2

[15] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020. 2

[16] Yuncheng Li, Jianchao Yang, Yale Song, Liangliang Cao, Jiebo Luo, and Li-Jia Li. Learning from noisy labels with distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1910–1918, 2017. 1

[17] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *Advances in neural information processing systems*, 33:20331–20342, 2020. 2

[18] Sheng Liu, Zhihui Zhu, Qing Qu, and Chong You. Robust training under label noise by over-parameterization. In *International Conference on Machine Learning*, pages 14153–14172. PMLR, 2022. 2

[19] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015. 2

[20] Kailang Ma, Yu Sun, Jian Cui, Dawei Li, Zhenyu Guan, and Jianwei Liu. Instance-wise batch label restoration via gradients in federated learning. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[21] Duc Tam Nguyen, Chaithanya Kumar Mummadi, Thi Phuong Nhung Ngo, Thi Hoai Phuong Nguyen, Laura Beggel, and Thomas Brox. Self: Learning to filter noisy labels with self-ensembling. *arXiv preprint arXiv:1910.01842*, 2019. 3

[22] Diane Oyen, Michal Kucer, Nicolas Hengartner, and Har Simrat Singh. Robustness to label noise depends on the shape of the noise distribution. *Advances in Neural Information Processing Systems*, 35:35645–35656, 2022. 2

[23] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 2017. 2

[24] Cosmin Octavian Pene, Amirmasoud Ghiassi, Taraneh Younesian, Robert Birke, and Lydia Y Chen. Multi-label gold asymmetric loss correction with single-label regulators. *arXiv preprint arXiv:2108.02032*, 2021. 2

[25] Hieu Pham, Zihang Dai, Qizhe Xie, and Quoc V Le. Meta pseudo labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11557–11568, 2021. 2

[26] Lars Schmarje, Monty Santarossa, Simon-Martin Schröder, Claudius Zelenka, Rainer Kiko, Jenny Stracke, Nina Volkmann, and Reinhard Koch. A data-centric approach for improving ambiguous labels with combined semi-supervised classification and clustering. In *European Conference on Computer Vision*, pages 363–380. Springer, 2022. 2

[27] Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. *Advances in neural information processing systems*, 32, 2019. 2, 5

[28] Jun Shu, Qian Zhao, Zongben Xu, and Deyu Meng. Meta transition adaptation for robust deep learning with noisy labels. *arXiv preprint arXiv:2006.05697*, 2020. 2

[29] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017. 1

[30] Mitchell Keren Taraday and Chaim Baskin. Enhanced meta label correction for coping with label corruption. *arXiv preprint arXiv:2305.12961*, 2023. 1, 2, 5

[31] Yuanpeng Tu, Boshen Zhang, Yuxi Li, Liang Liu, Jian Li, Yabiao Wang, Chengjie Wang, and Cai Rong Zhao. Learning from noisy labels with decoupled meta label purifier. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19934–19943, 2023. 1, 7, 8

[32] Zhen Wang, Guosheng Hu, and Qinghua Hu. Training noise-robust deep neural networks via meta-learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4524–4533, 2020. 1

[33] Jiaheng Wei, Hangyu Liu, Tongliang Liu, Gang Niu, Masashi Sugiyama, and Yang Liu. To smooth or not? when label smoothing meets noisy labels. *Learning*, 1(1):e1, 2021. 2

[34] Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. Learning to purify noisy labels via meta soft label corrector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10388–10396, 2021. 2, 3

[35] Yuhao Wu, Jiangchao Yao, Xiaobo Xia, Jun Yu, Ruxin Wang, Bo Han, and Tongliang Liu. Mitigating label noise on graph via topological sample selection. *arXiv preprint arXiv:2403.01942*, 2024. 2

[36] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2

[37] Xiaobo Xia, Tongliang Liu, Bo Han, Mingming Gong, Jun Yu, Gang Niu, and Masashi Sugiyama. Sample selection with uncertainty of losses for learning with noisy labels. *arXiv preprint arXiv:2106.00445*, 2021. 2

[38] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. Learning from massive noisy labeled data for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2691–2699, 2015. 7

[39] Youjiang Xu, Linchao Zhu, Lu Jiang, and Yi Yang. Faster meta update strategy for noise-robust deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 144–153, 2021. 2

[40] Shuo Yang, Erkun Yang, Bo Han, Yang Liu, Min Xu, Gang Niu, and Tongliang Liu. Estimating instance-dependent label-noise transition matrix using dnns. 2021. 1

[41] Jiangchao Yao, Hao Wu, Ya Zhang, Ivor W Tsang, and Jun Sun. Safeguarded dynamic label regression for noisy supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 9103–9110, 2019. 5

[42] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021. 1

[43] Weihe Zhang, Yali Wang, and Yu Qiao. Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7373–7382, 2019. 1, 2

[44] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, 31, 2018. 2

[45] Zizhao Zhang, Han Zhang, Sercan O Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9294–9303, 2020. 2

[46] Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. Meta label correction for noisy label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11053–11061, 2021. 1, 7

[47] Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with instance-dependent label noise. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10113–10123, 2021. 1, 5