LEARNING WITH NOISY LABELS BY EFFICIENT TRANSITION MATRIX ESTIMATION TO COMBAT LABEL MISCORRECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent studies on learning with noisy labels have shown remarkable performance by exploiting a small clean dataset. In particular, model agnostic meta-learning-based label correction methods further improve performance by correcting noisy labels on the fly. However, there is no safeguard on the label miscorrection, resulting in unavoidable performance degradation. Moreover, every training step requires at least three back-propagations, significantly slowing down the training speed. To mitigate these issues, we propose a robust and efficient meta-learning method that learns a label transition matrix on the fly. Employing the transition matrix makes the classifier skeptical about all the corrected samples, which alleviates the miscorrection issue. We also introduce a two-head architecture to efficiently learn the label transition matrix every iteration within a single back-propagation, so that the matrix estimate closely follows the shifting distribution induced by label correction. Extensive experiments demonstrate that our approach shows the best performance in training efficiency while having comparable or better accuracy than existing methods.

1 INTRODUCTION

In the last decade, supervised learning has achieved great success by leveraging an abundant amount of annotated data to solve various classification tasks such as image classification (He et al., 2016), object detection (Girshick et al., 2014), and face recognition (Taigman et al., 2014). It has been proven both theoretically and empirically that the performance of supervised learning-based classification models steadily improves as the size of annotated data increases (Goodfellow et al., 2016; Charikar et al., 2017; Floridi & Chiriatti, 2020). However, we cannot avoid *noisy labels* due to its coarse-grained annotation sources (Hendrycks et al., 2018; Zheng et al., 2021), resulting in performance degradation (Chen et al., 2019).

Many methods have been proposed to build a classifier that is robust to noisy labels. Unlike traditional methods (Mnih & Hinton, 2012; Van Rooyen et al., 2015; Azadi et al., 2015; Patrini et al., 2016) which assume that all the given labels are potentially corrupted, recently proposed methods utilize an inexpensively obtained small clean dataset to improve performance further. Based on the clean data set, loss correction methods (Hendrycks et al., 2018; Wang et al., 2020a) reduce the influence of noisy labels by modifying loss functions and re-weighting methods (Ren et al., 2018; Shu et al., 2019; Bahri et al., 2020; Ghosh & Lan, 2021) penalize samples which are likely to be noisy labels. Especially, recent label correction methods (Wu et al., 2020b; Zheng et al., 2021) achieve remarkable performance based on model-agnostic meta-learning (MAML) (Finn et al., 2017). These methods relabel noisy labels to directly reduce the noise level, raising the theoretical upper bound of the predictive performance (See Appendix A.1).

However, there are two challenges for these MAML-based label correction methods: (1) *The label correction methods blindly trust the already miscorrected labels*. Erroneously corrected labels are often kept throughout the training, which causes the model to learn the miscorrected labels as ground-truth labels. Several studies (Mirzasoleiman et al., 2020; Wu et al., 2020b) attempt to tackle this through training techniques such as soft labels, whereas it does not fundamentally solve the problem. (2) *MAML-based methods are inherently slow in training, resulting in excessive computational* *overhead*. The inefficiency comes from multiple training steps per single iteration of MAML-based methods, including virtual updates with inner optimization loops.

To alleviate these issues, we propose a robust and efficient meta-learning method called ***LT2L*** (**L**earning **T**ransition Matrix **to** **L**earn). LT2L efficiently learns a transition matrix to learn with noisy labels while continuously correcting them on-the-fly. It is theoretically proven that the correctly estimated label transition matrix is useful to obtain a statistically consistent classifier from noisy labels (Xia et al., 2019; Yao et al., 2020) (See Appendix A.2), i.e., more robust to noisy labels. To this end, we adopt a two-head architecture that consists of two classifiers, a noisy and a clean classifier, with a shared feature extractor. For every iteration, the noisy classifier estimates the label transition matrix shifted by the label correction. On the other hand, the clean classifier is trained to be statistically consistent by leveraging the estimated transition matrix. Using the output of the clean classifier, LT2L relabels noisy labels to reduce the noise level. Our proposed LT2L has a safeguard for the miscorrected labels since it adaptively estimates the transition matrix on every iteration, so that the clean classifier stays equally skeptical towards all the corrected labels. Furthermore, our efficient meta-learning method jointly optimizes the two-head architecture with only a single back-propagation for each iteration, boosting training speed.

Experimental results show that our method achieves state-of-the-art performance by a large margin on both the synthetic and real-world noisy label datasets, various noise levels of *CIFAR* (Krizhevsky et al., 2009) and *Clothing1M* (Xiao et al., 2015), respectively. We demonstrate the exceptional training speed of our proposed LT2L while achieving better performance compared to baselines. Finally, we conduct a thorough analysis to understand the inner mechanisms of our proposed method.

Our contribution in this paper is threefold:

- We propose a robust and efficient method that learns a transition matrix to learn with noisy labels while continuously correcting them on the fly. To the best of our knowledge, this is the first attempt to improve the label correction with the transition matrix estimation.
- Our proposed method boosts training speed by employing a two-head architecture so that the label transition matrix can be learned with a single back-propagation.
- Extensive experiments validate the efficacy of our proposed method in terms of both training speed and predictive performance.

2 Related Work

Traditional methods of Learning with Noisy Labels assume that labels in all the training samples are potentially corrupted. They can be categorized as follows: *various loss functions* (Natarajan et al., 2013; Van Rooyen et al., 2015; Patrini et al., 2016; Ghosh et al., 2017; Zhang & Sabuncu, 2018; Wang et al., 2019; Van Rooyen et al., 2015; Lukasik et al., 2020; Ma et al., 2020; Liu & Guo, 2020; Lienen & Hüllermeier, 2021), *regularizations* (Azadi et al., 2015; Jindal et al., 2016; Hendrycks et al., 2019a; Hu et al., 2019; Menon et al., 2020; Lukasik et al., 2020; Han et al., 2020b; Song et al., 2019b; Liu et al., 2020; Lienen & Hüllermeier, 2021; Cao et al., 2020; Hendrycks et al., 2019b; Nishi et al., 2021; Ortego et al., 2020; Li et al., 2020a; Zhang et al., 2017; Harutyunyan et al., 2020; Li et al., 2020b; Ma et al., 2018), *re-weighting training samples* (Ren et al., 2018; Jiang et al., 2018; Mirzasoleiman et al., 2020; Liu & Tao, 2015; Wang et al., 2017; Thulasidasan et al., 2019; Chen et al., 2019; Huang et al., 2020; Wang et al., 2020b; Wu et al., 2020a;a; Pleiss et al., 2020), and *correcting noisy labels* (Tanaka et al., 2018; Yi & Wu, 2019; Han et al., 2019; Song et al., 2019a; Zheng et al., 2020; Guo et al., 2020). However, different losses or regularizations yield inferior performance to state-of-the-art methods (Zhang et al., 2017; Mirzasoleiman et al., 2020; Hu et al., 2019; Li et al., 2020b), and re-weighting methods often filter out noisy but helpful samples for extracting features to show sub-optimal performance (Song et al., 2019a; Wu et al., 2020b; Zheng et al., 2021; Mirzasoleiman et al., 2020; Chang et al., 2017; Lin et al., 2017; Shrivastava et al., 2016). *Label correction* methods circumvent their shortcomings by relabeling so that the feature extractor leverage the corrected labels. However, label correction methods also have a limitation in that they are prone to propagate the error when miscorrected labels are continuously accumulated (Mirzasoleiman et al., 2020; Wu et al., 2020b; Zheng et al., 2021). Others *correct the training loss* by estimating a label transition matrix (Mnih & Hinton, 2012; Reed et al., 2014; Sukhbaatar et al., 2014; Bekker & Goldberger, 2016; Patrini et al., 2017; Goldberger & Ben-Reuven, 2016; Yao et al., 2019; Xia et al., 2019; Yao et al., 2020) to build a statistically consistent classifier, where the methods need multiple training stages; e.g., include a separate pretraining stage. In this paper, we join a simple label correction method with estimating the label transition matrix to alleviate the miscorrection issue caused by miscorrected noisy labels, which only requires a single training stage.

Learning with Noisy Label via Small Clean Dataset. Several recent studies argue that a small clean dataset is easily obtained; hence one can further devise a method that effectively leverages it. Many studies have successfully adapted the idea and shown massive performance improvement compared to the traditional methods. Early methods (Hendrycks et al., 2018; Bahri et al., 2020; Zhang et al., 2020) require multiple training stages where it hinders the training efficiency. Recent studies widely adopt MAML (Finn et al., 2017) to various strategies discussed above: sample re-weighting (Veit et al., 2017; Lee et al., 2018; Jiang et al., 2018; Ren et al., 2018; Li et al., 2019; Shu et al., 2019), label correction (Wu et al., 2020b; Zheng et al., 2021), and label transition matrix estimation (Wang et al., 2020a). These approaches first perform a virtual update with the noisy dataset, find optimal parameters using the clean dataset, and update the actual parameters by the found parameters. This virtual update process requires three back-propagations per iteration, leading to at least three times the computational cost. Our proposed label correction method learns the label transition matrix with a single back-propagation, greatly enhancing the training speed while showing comparable or better performance to existing state-of-the-art methods. Additional related works are described in Appendix D.

3 METHODOLOGY

Existing label correction methods try to find and fix noisy labels to utilize them as clean samples in model training, where they can improve the classification performance by reducing the noise level of the whole training samples. However, erroneously corrected samples, i.e., clean samples deemed noisy, or vice versa, are often kept throughout the model training. Since current label correction methods blindly trust these miscorrected labels, this behavior degrades the classification performance under the noisy label situation (§ 4.4).

In this section, we show that the accurately estimated label transition matrix with the clean dataset alleviates the miscorrection problem of existing label correction methods. Further, we describe our efficient meta-learning method estimating label transition matrix for every training iteration while correcting noise labels. Our proposed method is summarized in Algorithm 1.

3.1 BATCH FORMATION

In few-shot learning studies (Finn et al., 2017; Snell et al., 2017; Sung et al., 2018), meta-learning often address the episodic batch formation, a nontrivial sampling method where each mini-batch is composed as a few-shot task. We closely follow the previous works to ensure the effective estimation of the label transition matrix, where it benefits from having a certain amount of clean samples. In our method, the task is defined as estimating the transition matrix representing the shifted noisy label distribution caused by label correction, using a clean batch. We first compose a clean batch d with randomly chosen K samples for the entire N classes in the clean dataset $\mathcal{D}$[1]. Note tha,t unlike our method, other approaches based on meta-learning (Zheng et al., 2021; Wu et al., 2020b; Shu et al., 2019; Ren et al., 2018) randomly compose the clean batch, where the noisy batch $\bar{d}$ is randomly sampled from the noisy dataset $\bar{\mathcal{D}}$ similar to other works. Namely, we have the clean batch $d = \{(x_i, y_i)\}_{i=1}^{KN}$ and the noisy batch $\bar{d} = \{(x_i, \bar{y}_i)\}_{i=1}^{M}$, where x is an input, $y, \bar{y} \in \mathbb{R}^N$ are the labels of x, and M is the size of the noisy batch which we set as $M = KN$ for simplicity.

3.2 TRANSITION MATRIX ESTIMATION

Each element T_{ij} of the label transition matrix $\boldsymbol{T} \in \mathbb{R}^{N \times N}$ is defined as the probability of a clean label i to be corrupted as a noisy label j, i.e. $T_{ij} = p(\bar{y} = j | y = i)$. It is well-known that a robust classifier can be obtained with the accurately estimated label transition matrix (Sukhbaatar et al., 2014; Patrini et al., 2017; Hendrycks et al., 2018; Xia et al., 2019; Yao et al., 2020). We choose a

[1]Similar to N-way K-shot few-shot learning in Finn et al. (2017); Vinyals et al. (2016); Snell et al. (2017); Sung et al. (2018).

Algorithm 1 **L**earning **T**ransition Matrix **to L**earn (**LT2L**)

Input: Clean dataset $\mathcal{D}$, noisy dataset $\bar{\mathcal{D}}$.
Hyper-parameters: Label correction threshold ρ, Controllable loss ratio for noisy classifier λ.
Output: Clean classifier $f_{\phi,\theta}$ where linear classifier θ and feature extractor ϕ.
Randomly initialize common feature extractor ϕ.
Randomly initialize linear classifiers θ and $\bar{\theta}$ for clean labels and noisy labels, respectively.
for each epoch $i = 0, \cdots$ **do**
  for each iteration in epoch i **do**
    Sample mini-batch $d \sim \mathcal{D}$, $\bar{d} \sim \bar{\mathcal{D}}$.
    $\hat{T} \leftarrow \left(\sum_{(x,y)\in d} y f_{\phi,\bar{\theta}}(x)^\top\right) \text{diag}^{-1}\left(\sum_{(x,y)\in d} y\right)$
    $J \leftarrow \sum_{(x,y)\in d} \mathcal{L}\left(f_{\phi,\theta}(x), y\right) + \sum_{(x,\bar{y})\in \bar{d}} \left(\mathcal{L}\left(\hat{T}^\top f_{\phi,\theta}(x), \bar{y}\right) + \lambda \mathcal{L}\left(f_{\phi,\bar{\theta}}(x), \bar{y}\right)\right)$
    $\bar{\mathcal{D}} \leftarrow (\bar{\mathcal{D}} - \bar{d}) \cup \left\{ \left(x, \begin{cases} \bar{y}^*, & \text{if } \max(f_{\phi,\theta}(x)) < \rho \\ \lfloor f_{\phi,\theta}(x)/\max(f_{\phi,\theta}(x)) \rfloor, & \text{otherwise} \end{cases} \right) \middle| (x, \bar{y}) \in \bar{d} \right\}$
    Update $\phi, \theta, \bar{\theta}$ using $\nabla_{\phi,\theta,\bar{\theta}} J$ with a single back-propagation.
  end for
end for

simple but accurate method that directly estimates the posterior with a clean dataset (Hendrycks et al., 2018; Xia et al., 2019; Yao et al., 2020), where there are other methods that estimate the label transition matrix (Mnih & Hinton, 2012; Reed et al., 2014; Sukhbaatar et al., 2014; Bekker & Goldberger, 2016; Goldberger & Ben-Reuven, 2016; Patrini et al., 2017). Following the assumption of the previous work (Hendrycks et al., 2018; Chen et al., 2020; Berthon et al., 2020; Xia et al., 2020), we also assume conditional independence of $\bar{y}$ and y given x.

$$p(\bar{y}|y,x) = p(\bar{y}|y) \int p(x|\bar{y},y)dx = \int p(\bar{y}|y,x)p(x|y)dx = \int p(\bar{y}|x)p(x|y)dx. \quad (1)$$

By parameterizing a feature extractor $\bar{\phi}$ and a linear classifier $\bar{\theta}$, we obtain $p(\bar{y}|x) = f_{\bar{\phi},\bar{\theta}}(x)$ where $f_{\bar{\phi},\bar{\theta}}$ is the noisy classifier that consists of the linear classifier and the feature extractor trained only with the noisy labels. If the noisy classifier $f_{\bar{\phi},\bar{\theta}}$ gives a perfect prediction for the noisy data, we can estimate the transition probability $p(\bar{y}|y)$ using the clean samples $(x, y) \in d$ as follows (See Appendix A.3 for the mathematical details):

$$\widehat{\boldsymbol{T}} \leftarrow \left(\sum_{(x,y)\in d} y f_{\bar{\phi},\bar{\theta}}(x)^\top\right) \text{diag}^{-1}\left(\sum_{(x,y)\in d} y\right). \quad (2)$$

We emphasize the importance of the transition matrix estimation, as its accuracy determines the bounds of the generalization error of the classifier (Xia et al., 2019). However, the limited number of clean samples inside a single batch may yield an inaccurate transition matrix, even with the ideal $f_{\bar{\phi},\bar{\theta}}$. We analyze the upper bound of the estimation error as follows:

Theorem 1. *Assume the Frobenius norm of the weight matrices $\bar{\phi}_1, ..., \bar{\phi}_{H-1}, \bar{\theta}$ are at most $\bar{\Phi}_1, ..., \bar{\Phi}_{H-1}, \bar{\Theta}$ for H-layer neural networks $f_{\bar{\phi},\bar{\theta}}$. Let the loss function be L-Lipschitz continuous w.r.t. $f_{\bar{\phi},\bar{\theta}}$. Let the activation functions be 1-Lipschitz, positive-homogeneous, and applied element-wise (such as ReLU). Let x be upper bounded by B, i.e., for any $x \in \mathcal{X}$, $\|x\| \leq B$. Then, for $\epsilon \geq 0$*

$$p\left(\left|\widehat{T}_{ij} - T_{ij}\right| > \epsilon\right) \leq \frac{NLB(\sqrt{2H\log 2}+1)\bar{\Theta}\Pi_{h=1}^{H-1}\bar{\Phi}_i + \sqrt{1/2\log(1/\epsilon)}}{\sqrt{|\bar{\mathcal{D}}|}} + 2\exp\left(-2\epsilon^2 K\right). \quad (3)$$

Proof. See Appendix A.4. ☐

Although the upper bound of the estimation error of the transition matrix is affected by the batch size K, we empirically verify that small K does not necessarily harm the classification performance (See Appendix C.6.2).

3.3 LEARNING WITH ESTIMATED TRANSITION MATRIX

A clean classifier $f_{\phi,\theta}$ is trained with the estimated transition matrix $\widehat{\boldsymbol{T}}$:

$$\arg\min_{\phi,\theta} \sum_{(x,y)\in d} \mathcal{L}\left(f_{\phi,\theta}(x), y\right) + \sum_{(x,\bar{y})\in \bar{d}} \mathcal{L}\left(\widehat{\boldsymbol{T}}^{\top} f_{\phi,\theta}(x), \bar{y}\right), \tag{4}$$

given the cross-entropy loss function $\mathcal{L}$, where the feature extractor ϕ and the linear classifier θ form the clean classifier $f_{\phi,\theta}$ which estimates clean labels. If $\widehat{\boldsymbol{T}}$ is correctly estimated, the clean classifier $f_{\phi,\theta}$ becomes statistically consistent (Sukhbaatar et al., 2014; Patrini et al., 2017; Hendrycks et al., 2018; Xia et al., 2019; Yao et al., 2020). This approach makes the clean classifier skeptical towards corrected labels, hence avoiding the miscorrection issue.

On the other hand, the noisy classifier $f_{\bar{\phi},\bar{\theta}}$ is trained to model the noisy label distribution.

$$\arg\min_{\bar{\phi},\bar{\theta}} \sum_{(x,\bar{y})\in \bar{d}} \mathcal{L}\left(f_{\bar{\phi},\bar{\theta}}(x), \bar{y}\right) \tag{5}$$

We emphasize that updating the noisy classifier $f_{\bar{\phi},\bar{\theta}}$ every iteration is critical as it can adaptively model the ever-changing noisy label distribution on the fly, where the distribution constantly shifts as the noisy labels are actively corrected to reduce the noise level (See § 3.5).

3.4 EFFICIENT TRAINING

Similar to Vinyals et al. (2016); Jiang et al. (2018), we propose an efficient training scheme through weight sharing via two-head architecture. Where the architecture closely resembles the ones of Vinyals et al. (2016); Jiang et al. (2018), our two-head architecture only shares the feature extractor $\phi = \bar{\phi}$. Unlike the shared feature extractor, our architecture does not share the linear classifier since modeling both noisy and clean data distribution with a single linear classifier is impractical. Thus, we define the clean and noisy classifier as $f_{\phi,\theta}$ and $f_{\phi,\bar{\theta}}$, respectively, to produce our final objective function:

$$\arg\min_{\phi,\theta,\bar{\theta}} \sum_{(x,y)\in d} \mathcal{L}\left(f_{\phi,\theta}(x), y\right) + \sum_{(x,\bar{y})\in \bar{d}} \left(\mathcal{L}\left(\widehat{\boldsymbol{T}}^{\top} f_{\phi,\theta}(x), \bar{y}\right) + \lambda \mathcal{L}\left(f_{\phi,\bar{\theta}}(x), \bar{y}\right)\right) \tag{6}$$

where λ is a loss balancing factor. In order to prevent over-fitting on $\bar{d}$, we introduce λ to the final objective function. We search for the optimal hyperparameter λ for all of our experiments (See Appendix C.6.3).

Efficiency Analysis. Compared to the vanilla training scheme, which assumes that all labels are clean, we only add a single linear classifier $\bar{\theta}$ with only N additional parameters. Also, our loss only requires a single back-propagation, where the added linear classifier has a negligible computational burden. Our training scheme stands out even more compared to the existing MAML-based methods (Wu et al., 2020b; Zheng et al., 2021) or multi-stage training (Hendrycks et al., 2018; Bahri et al., 2020) (See § 4.2 and Figure 1).

3.5 LABEL CORRECTION

In this paper, we focus on the efficient, on-the-fly estimation of the label transition matrix to combat label miscorrection. To further demonstrate the effectiveness of our method, we employ a naïve label correction strategy where we feed each noisy set sample $x \in \bar{d}$ to the clean classifier $f_{\phi,\theta}$ to produce a probability vector. If the maximum probability $\max(f_{\phi,\theta}(x))$ is bigger than the threshold ρ, we correct its label to a more probable label. This strategy relies only on the most recent prediction of the model mid-training, so the decision is prone to change. Formally, we can describe as follows:

$$\begin{cases} \bar{y}^*, & \text{if } \max(f_{\phi,\theta}(x)) < \rho \\ \lfloor f_{\phi,\theta}(x) / \max(f_{\phi,\theta}(x)) \rfloor, & \text{otherwise} \end{cases} \tag{7}$$

where $\lfloor\cdot\rfloor$ denotes floor function and $\bar{y}^*$ denotes the original label from $\bar{d}$. Even with this simple strategy, our model shows better performance compared to the state-of-the-art methods. The experimental results suggest that replacing this strategy may further improve the model performance.

Table 1: Performance comparison on CIFAR-10/100 datasets under various noise level. Test accuracy (%) with 95% confidence interval of 5-runs is provided.

	Method	Symmetric Noise Level 20%	40%	60%	80%	Asymmetric Noise Level 20%	40%
CIFAR-10	L2RW	88.26 ± 0.79	83.76 ± 0.54	74.54 ± 1.54	42.60 ± 1.71	88.79 ± 0.63	85.86 ± 0.87
	MW-Net	89.76 ± 0.31	86.52 ± 0.28	81.68 ± 0.25	56.56 ± 3.07	91.31 ± 0.25	88.69 ± 0.37
	Deep kNN	90.02 ± 0.35	87.27 ± 0.39	82.80 ± 0.55	68.30 ± 1.21	89.97 ± 0.48	84.56 ± 0.87
	GLC	89.66 ± 0.10	85.30 ± 0.73	80.34 ± 0.73	67.44 ± 1.50	91.56 ± 0.66	**89.76 ± 0.89**
	MLoC	90.50 ± 0.71	87.20 ± 0.35	81.95 ± 0.44	54.64 ± 4.04	91.15 ± 0.16	89.35 ± 0.45
	MLaC	89.75 ± 0.62	86.63 ± 0.56	82.20 ± 0.81	71.94 ± 2.22	91.45 ± 0.32	**90.26 ± 0.48**
	MSLC	90.94 ± 0.45	88.36 ± 0.80	83.93 ± 1.21	64.90 ± 4.84	**91.45 ± 1.35**	89.26 ± 0.52
	LT2L (ours.)	**91.94 ± 0.28**	**90.07 ± 0.17**	**86.78 ± 0.31**	**79.52 ± 0.78**	**92.29 ± 0.10**	**90.43 ± 0.31**
CIFAR-100	L2RW	57.79 ± 1.88	44.82 ± 4.30	30.01 ± 1.74	10.71 ± 1.79	59.11 ± 2.74	55.12 ± 3.40
	MW-Net	66.73 ± 0.78	59.44 ± 0.91	49.19 ± 1.57	19.04 ± 1.21	67.90 ± 0.78	64.50 ± 0.34
	Deep kNN	59.60 ± 0.97	52.48 ± 1.37	39.90 ± 0.60	23.39 ± 0.75	57.71 ± 0.47	50.23 ± 1.12
	GLC	60.99 ± 0.64	49.00 ± 4.33	33.38 ± 4.09	20.38 ± 1.35	64.43 ± 0.43	54.20 ± 0.86
	MLoC	**68.16 ± 0.41**	62.09 ± 0.33	**54.49 ± 0.92**	20.23 ± 1.86	69.20 ± 0.59	66.48 ± 0.56
	MLaC	49.81 ± 5.59	35.15 ± 5.75	20.15 ± 2.81	12.85 ± 0.87	56.46 ± 3.54	49.20 ± 3.23
	MSLC	**68.62 ± 0.60**	**63.30 ± 0.49**	53.83 ± 0.70	21.07 ± 5.20	**70.86 ± 0.30**	**66.99 ± 0.69**
	LT2L (ours.)	**68.75 ± 0.60**	**63.82 ± 0.33**	**55.22 ± 0.64**	**37.36 ± 1.15**	**70.35 ± 0.51**	**67.93 ± 0.53**

4 EXPERIMENTS

In this section, we evaluate our proposed learning method, LT2L, in terms of predictive performance (§ 4.1) and efficiency (§ 4.2). We also validate the label correction performance to demonstrate that our method is better in correcting noisy labels (§ 4.3 and Appendix C.5) and experimentally show the robustness of our proposed method towards miscorrected labels. (§ 4.4). We further analyze whether our method successfully estimates the label transition matrix where the label correction shifts the true label transition matrix (§ 4.5 and Appendix C.4.3). Additional experimental results and further analyses are described in Appendix C. We provide the source codes[2] for the reproduction of the experiments conducted in this paper.

Baselines. We deliberately choose the baselines to encompass the three types of approaches in learning with noisy labels. *Re-weighting*: **L2RW** (Ren et al., 2018) learns to assign weights to training samples based on their gradients. **MW-Net** (Shu et al., 2019) trains an explicit weighting function with the training samples. **Deep kNN** (Bahri et al., 2020) applies the k-nearest neighbor algorithm to the logit layer of classifiers to find noisy samples. *Label transition matrix estimation*: **GLC** (Hendrycks et al., 2018) estimates the label transition matrix using the small clean dataset. **MLoC** (Zheng et al., 2021) considers the label transition matrix as trainable parameters to be obtained through meta-learning. *Label correction*: **MLaC** (Zheng et al., 2021) trains a label correction network as a meta-process to provide corrected labels. **MSLC** (Wu et al., 2020b) uses soft labels with loss balancing weight through meta-gradient descent step under the guidance of the clean dataset.

4.1 PREDICTIVE PERFORMANCE COMPARISON

4.1.1 CIFAR-10/100 WITH SYNTHETIC NOISE

Setup. CIFAR-10/100 (Krizhevsky et al., 2009) have been widely adopted to assess the robustness of the methods to noisy labels. Since CIFAR-10/100 are known as clean datasets, labels are synthetically manipulated to contain noisy labels, injecting two types of noise: symmetric and asymmetric. **Symmetric:** The labels are randomly flipped with uniform distribution. **Asymmetric:** the labels are flipped with class-dependent distribution, following the evaluation protocol of Patrini et al. (2017); Yao et al. (2019). We claim that most studies report the performance highly overfitted to the test set without hyperparameter tuning on the validation set (Wu et al., 2020b; Li et al., 2020a; Nishi et al., 2021; Ortego et al., 2020). Moreover, baseline models employ different backbone networks, making it challenging to dissect the performance improvement whether it originated from each method or the

[2] `https://anonymous.4open.science/r/LT2L-6014`

Table 2: Test accuracy (%) comparison on Clothing1M dataset with real-world label noise. The results except L2RW Ren et al. (2018) are taken from original papers.

Method	L2RW	MW-Net	GLC	MLoC	MLaC	MSLC	LT2L w/o LC	**LT2L (ours.)**
Accuracy (%)	72.04 ± 0.24	73.72	73.69	71.10	75.78	74.02	77.07 ± 0.52	**77.83 ± 0.17**

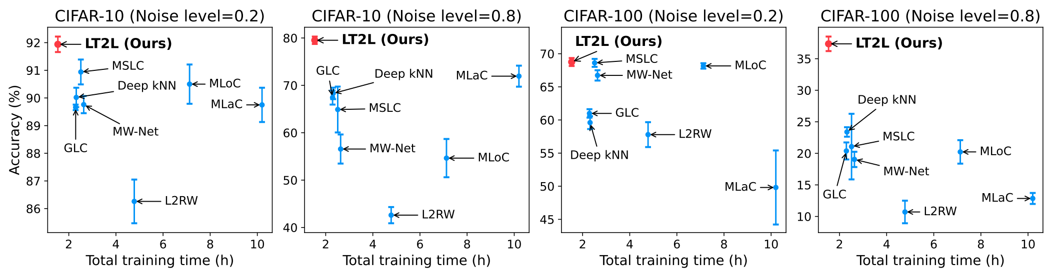

Figure 1: Plotting accuracy (%) **(y-axis)** according to total training time (hours) **(x-axis)**. Our proposed LT2L shows the best performance in training efficiency while having comparable or better accuracy.

backbone networks. Therefore, we first extract 5K samples as the validation set from the training set containing 50K samples and further extract 1K samples as the clean dataset. Then, we unify the backbone network as ResNet-34 (He et al., 2016), which is widely adopted in various baselines (Wu et al., 2020b; Liu et al., 2020). Note that we do our best to maintain the experimental settings of each method, including the hyperparameters written in the original paper. Detailed settings are deferred to Appendix B.

Results. Table 1 summarizes the evaluation results on CIFAR-10/100. For both CIFAR-10/-100, our proposed LT2L achieves state-of-the-art performance on various noise levels within 95% confidence intervals. Especially, under a high noise level (80%), our LT2L considerably outperforms the baselines with small variance on performance, which implies the robustness of our method (Li et al., 2016; 2017). These results demonstrate that our proposed method performs well in learning with noisy labels, especially considering its training efficiency (See § 4.2).

4.1.2 Clothing1M with Real-world Noise

Setup. Clothing1M (Xiao et al., 2015) is a noisy real-world dataset that consists of one million samples with additional 47K human-annotated clean samples. We use its original splits of clean and noise data. For a fair comparison, we employ ResNet-50 architecture pretrained with the ImageNet dataset (Deng et al., 2009) for the initial backbone architecture. Evaluation results on Clothing1M are summarized in Table 2. Except for L2RW that was not evaluated on Clothing1M, we borrow the reported performance of each baseline from its original paper.

Results. As shown in Table 2, our proposed LT2L achieves state-of-the-art performance on Clothing1M, beating the baselines by a large margin. This evaluation result indicates that our proposed LT2L is more applicable in real-world problems where label corruption frequently occurs.

4.2 Training Time Comparison

To verify the efficiency of our proposed LT2L, we compare it with the baselines in terms of accuracy by total training time. Total training time is measured on CIFAR-10/-100, respectively, with a single RTX 2080Ti GPU. Test accuracy shows the predictive performance on CIFAR-10/100 with 20% and 80% symmetric noise ratios, the mildest and most severe noise conditions, respectively. The detailed information for both iteration time and total training time is summarized in Appendix C.2. As shown in Figure 1, our proposed method, which learns the label transition matrix with the single back-propagation, makes model training more efficient than other baselines while showing better performance.

Table 3: Label correction performance comparison on CIFAR-10 with symmetric 80% noise. Accuracy (%) and Negative Log Likelihood (NLL) loss are calculated using the true labels before the synthetic noise is injected. Performance of the trained model on all training samples (Overall) and incorrectly labeled training samples (Incorrect) is measured. † denotes performance extracted from the meta model.

Method		L2RW	MW-Net	Deep kNN	GLC	MLoC	MLaC	MLaC†	MSLC	MSLC†	LT2L w/o LC	**LT2L (ours.)**
Acc.	Overall	0.4450	0.6024	0.6471	0.6900	0.6261	0.7567	0.7672	0.6762	0.2821	0.7559	**0.7847**
	Incorrect	0.4447	0.6024	0.6483	0.6903	0.6257	0.7569	0.7382	0.6755	0.2836	0.7560	**0.7861**
NLL	Overall	1.6684	1.6961	1.6085	1.3904	1.7492	0.9868	1.7004	1.2694	1.5989	1.0057	**0.8889**
	Incorrect	1.6674	1.6957	1.6084	1.3881	1.7493	0.9851	1.7299	1.2722	1.5990	1.0033	**0.8877**

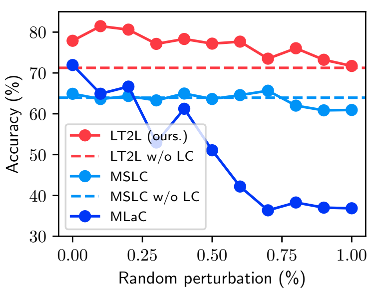

Figure 2: Robustness to miscorrected labels on CIFAR-10 with various perturbation strength. Test accuracy (%) of baselines and baselines without the label correction is provided.

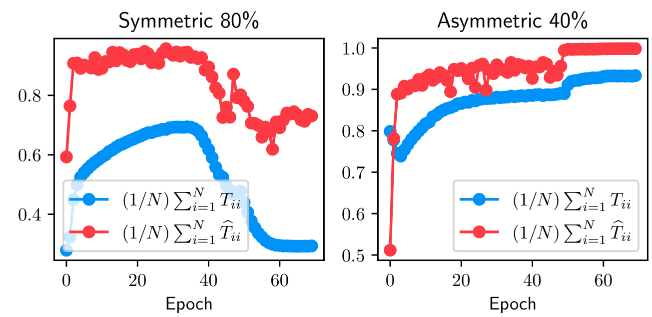

Figure 3: The plot for the mean of the diagonal term in true transition matrix $\boldsymbol{T}$ and our estimated transition matrix $\widehat{\boldsymbol{T}}$ according to the epoch on CIFAR-10 dataset with symmetric 80% and asymmetric 40% noise.

4.3 Label Correction Performance Comparison

We analyze the predictive performance of the baseline methods on all the training samples (Overall) and the wrongly labeled subset of them (Incorrect), respectively. Our method can successfully correct the noisy labels, where using the label correction further improves the correction performance. This implies that our LT2L may be helpful in further cleansing the noisy training set. We also compare the performance between Overall and Incorrect cases. Re-weighting (L2RW, MW-Net, Deep kNN) and transition matrix estimation-based methods (GLC, MLoC) show similar performance between the two. However, the performance of the meta-model of MLaC is worse for the Incorrect case, which indicates that the correction from the meta-model is less effective where the labels are wrong. Also, notable underperformance of the meta-model of MSLC may indicate the inefficacy of the meta-model. We also analyze the meta-model of the re-weighting methods in the Appendix C.5, where they do not distinguish the wrongly labeled samples well.

4.4 Robustness to Miscorrection: What Happens if Labels are Wrongly Corrected?

This subsection illustrates the robustness of our label correction method to miscorrected labels by comparing it with other label correction methods (MLaC and MSLC) which blindly trust the miscorrected labels as the ground-truth, where we verify the imperfect corrections (See § 4.3). We examine how much this behavior deteriorates the predictive performance.

Setup. We experiment on CIFAR-10 with symmetric 80% noise where there are a maximum number of noisy labels to correct. To simulate the miscorrection, we perturb the corrected labels by injecting artificial noise. We control the degree of random perturbation to observe the robustness of each method on various levels of miscorrection. We further assess the robustness of our LT2L and MSLC by comparing it with the performance obtained without label correction.

Results. As shown in Figure 2, our proposed LT2L outperforms MLaC and MSLC on all the degrees of the random perturbation. MLaC shows steep performance degradation when perturbation worsens, i.e., there are more miscorrected labels. This observation reveals the susceptibility of MLaC. MSLC shows trivial performance gains when labels are corrected, implying that it is not using the full benefits of label correction. Furthermore, when highly perturbed, MSLC performance worsens if it attempts to correct the labels. In contrast, the label correction of our LT2L improves performance even in harsh situations. LT2L does not degrade performance even if the correction becomes useless (100% perturbation), and we argue that this is due to the safeguard provided by the transition matrix. These observations demonstrate that our LT2L builds a more robust classifier to miscorrected labels through its efficient estimation of the label transition matrix, and it could act as a safeguard combating the miscorrected labels.

4.5 On-the-fly Estimation of the Label Transition Matrix

Our proposed LT2L newly estimates the label transition matrix on every iteration, where the matrix is constantly shifted by the label correction. To assess the quality of the estimated transition matrix, we compare it with the true label transition matrix.

Setup. We train LT2L on CIFAR-10 with symmetric 80% and asymmetric 40% noise, which are harsh conditions on symmetric and asymmetric noise injection, respectively. We compare the estimated label transition matrix $\widehat{\boldsymbol{T}}$ with the true label transition matrix $\boldsymbol{T}$ by observing the mean of diagonal term values for each epoch. The mean of the diagonal term in the transition matrix represents the average of the probability that a sample is mapped to a clean label.

Results. Figure 3 shows the overall tendency of the estimated transition matrix (red plots) to follow the true label matrix (blue plots). In the asymmetric 40% setting, the mean of diagonal term values of the true label transition matrix $\boldsymbol{T}$ gradually increases (blue plots), which indicates the dataset is cleansed by the label correction. However, in the symmetric 80% case, the mean of diagonal term values of the oracle transition matrix $\boldsymbol{T}$ decreases at the middle of the training. As we maintain the fixed threshold ρ, the total number of corrected samples decreases. Nonetheless, we can conclude that the transition matrix is successfully estimated on shifting noise levels.

Additionally, we observe that the estimated transition matrix $\widehat{\boldsymbol{T}}$ shows higher mean values, i.e., being overconfident on the clean dataset samples. Theoretically, $f_{\phi,\bar{\theta}}$ should correctly approximate the noisy label distribution given enough number of clean samples (See Appendix A.4), but it seems to be overfitting to the clean dataset in practice. This observation is consistent with the popular belief that deep neural networks tend to learn clean samples first and noisy samples later (Arpit et al., 2017). For the better estimation of the transition matrix to yield a more robust classifier (Han et al., 2020a; Xia et al., 2019; Mirzasoleiman et al., 2020; Yao et al., 2020), it appears that we need to address the overfitting through additional components.

5 Conclusion

In this work, we propose a meta-learning-based method, LT2L, which efficiently learns a label transition matrix that mitigates the label miscorrection problem of existing label correction methods. Our proposed LT2L accurately estimates the label transition matrix using a small clean dataset even if the samples are miscorrected. Moreover, our LT2L is highly efficient compared to existing methods since it requires single back-propagation through two-head architecture. Extensive experiments show that our method is the fastest and the most robust classifier. Especially, our method achieves the state-of-the-art performance on both the real-world noise dataset (Clothing1M) and the synthetic dataset on various noise levels (CIFAR). The detailed analysis shows that our method is robust to miscorrected labels by efficiently estimating the transition matrix shifted by the label correction.

REFERENCES

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

Samaneh Azadi, Jiashi Feng, Stefanie Jegelka, and Trevor Darrell. Auxiliary image regularization for deep cnns with noisy labels. *arXiv preprint arXiv:1511.07069*, 2015.

Dara Bahri, Heinrich Jiang, and Maya Gupta. Deep k-nn for noisy labels. In *International Conference on Machine Learning*, pp. 540–550. PMLR, 2020.

Peter Bartlett, Dylan J Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1706.08498*, 2017.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

Alan Joseph Bekker and Jacob Goldberger. Training deep neural-networks based on unreliable labels. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2682–2686. IEEE, 2016.

Antonin Berthon, Bo Han, Gang Niu, Tongliang Liu, and Masashi Sugiyama. Confidence scores make instance-dependent label-noise learning possible. *arXiv preprint arXiv:2001.03772*, 2020.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

Kaidi Cao, Yining Chen, Junwei Lu, Nikos Arechiga, Adrien Gaidon, and Tengyu Ma. Heteroskedastic and imbalanced deep learning with adaptive regularization. *arXiv preprint arXiv:2006.15766*, 2020.

Haw-Shiuan Chang, Erik Learned-Miller, and Andrew McCallum. Active bias: Training more accurate neural networks by emphasizing high variance samples. *arXiv preprint arXiv:1704.07433*, 2017.

Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 47–60, 2017.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

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

Pengfei Chen, Junjie Ye, Guangyong Chen, Jingwei Zhao, and Pheng-Ann Heng. Robustness of accuracy metric and its inspirations in learning with noisy labels. *arXiv preprint arXiv:2012.04193*, 2020.

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

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

Amnon Drory, Shai Avidan, and Raja Giryes. How do neural networks overcome label noise. *arXiv Preprint*, 2018.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135. PMLR, 2017.

Luciano Floridi and Massimo Chiriatti. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30(4):681–694, 2020.

Aritra Ghosh and Andrew Lan. Do we really need gold samples for sample weighting under label noise? In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3922–3931, 2021.

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

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

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

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. Size-independent sample complexity of neural networks. In *Conference On Learning Theory*, pp. 297–299. PMLR, 2018.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. *arXiv preprint arXiv:1801.08930*, 2018.

Melody Guan, Varun Gulshan, Andrew Dai, and Geoffrey Hinton. Who said what: Modeling individual labelers improves classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pp. 1321–1330. PMLR, 2017.

Jiaxian Guo, Mingming Gong, Tongliang Liu, Kun Zhang, and Dacheng Tao. Ltf: A label transformation framework for correcting label shift. In *International Conference on Machine Learning*, pp. 3843–3853. PMLR, 2020.

Bo Han, Jiangchao Yao, Gang Niu, Mingyuan Zhou, Ivor Tsang, Ya Zhang, and Masashi Sugiyama. Masking: A new perspective of noisy supervision. *arXiv preprint arXiv:1805.08193*, 2018a.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *arXiv preprint arXiv:1804.06872*, 2018b.

Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor Tsang, and Masashi Sugiyama. Sigua: Forgetting may make learning with noisy labels more robust. In *International Conference on Machine Learning*, pp. 4006–4016. PMLR, 2020a.

Jiangfan Han, Ping Luo, and Xiaogang Wang. Deep self-learning from noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5138–5147, 2019.

Kai Han, Yunhe Wang, Yixing Xu, Chunjing Xu, Enhua Wu, and Chang Xu. Training binary neural networks through learning with noisy supervision. In *International Conference on Machine Learning*, pp. 4017–4026. PMLR, 2020b.

Hrayr Harutyunyan, Kyle Reing, Greg Ver Steeg, and Aram Galstyan. Improving generalization by controlling label-noise information in neural network weights. In *International Conference on Machine Learning*, pp. 4071–4081. PMLR, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. *arXiv preprint arXiv:1802.05300*, 2018.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pp. 2712–2721. PMLR, 2019a.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *arXiv preprint arXiv:1906.12340*, 2019b.

Youngkyu Hong, Seungju Han, Kwanghee Choi, Seokjun Seo, Beomsu Kim, and Buru Chang. Disentangling label distribution for long-tailed visual recognition. *arXiv preprint arXiv:2012.00321*, 2020.

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. *arXiv preprint arXiv:1905.11368*, 2019.

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

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

Ishan Jindal, Matthew Nokleby, and Xuewen Chen. Learning deep networks from noisy labels with dropout regularization. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 967–972. IEEE, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5447–5456, 2018.

Yoonho Lee and Seungjin Choi. Gradient-based meta-learning with learned layerwise metric and subspace. In *International Conference on Machine Learning*, pp. 2927–2936. PMLR, 2018.

Dongsheng Li, Chao Chen, Qin Lv, Junchi Yan, Li Shang, and Stephen Chu. Low-rank matrix approximation with stability. In *International Conference on Machine Learning*, pp. 295–303. PMLR, 2016.

Dongsheng Li, Chao Chen, Wei Liu, Tun Lu, Ning Gu, and Stephen M Chu. Mixture-rank matrix approximation for collaborative filtering. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 477–485, 2017.

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*, pp. 5051–5059, 2019.

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

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

Julian Lienen and Eyke Hüllermeier. From label smoothing to label relaxation. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence, AAAI, Online, February 2-9, 2021*. AAAI Press, 2021.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. Early-learning regularization prevents memorization of noisy labels. *arXiv preprint arXiv:2007.00151*, 2020.

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

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

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *International Conference on Machine Learning*, pp. 6448–6458. PMLR, 2020.

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

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

Aditya Krishna Menon, Brendan Van Rooyen, and Nagarajan Natarajan. Learning from binary labels with instance-dependent corruption. *arXiv preprint arXiv:1605.00751*, 2016.

Aditya Krishna Menon, Ankit Singh Rawat, Sashank J Reddi, and Sanjiv Kumar. Can gradient clipping mitigate label noise? 2020.

Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. Coresets for robust training of deep neural networks against noisy labels. *Advances in Neural Information Processing Systems*, 33, 2020.

Volodymyr Mnih and Geoffrey E Hinton. Learning to label aerial images from noisy data. In *Proceedings of the 29th International conference on machine learning (ICML-12)*, pp. 567–574, 2012.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Stephen J Montgomery-Smith. The distribution of rademacher sums. *Proceedings of the American Mathematical Society*, 109(2):517–522, 1990.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *NIPS*, volume 26, pp. 1196–1204, 2013.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. *arXiv preprint arXiv:1707.09564*, 2017.

Kento Nishi, Yi Ding, Alex Rich, and Tobias Höllerer. Augmentation strategies for learning with noisy labels. *arXiv preprint arXiv:2103.02130*, 2021.

Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Multi-objective interpolation training for robustness to label noise. *arXiv preprint arXiv:2012.04462*, 2020.

Giorgio Patrini, Frank Nielsen, Richard Nock, and Marcello Carioni. Loss factorization, weakly supervised learning and label noise robustness. In *International conference on machine learning*, pp. 708–717. PMLR, 2016.

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

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

Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343. PMLR, 2018.

Filipe Rodrigues and Francisco Pereira. Deep learning from crowds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Clayton Scott et al. Calibrated asymmetric surrogate losses. *Electronic Journal of Statistics*, 6: 958–992, 2012.

Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 761–769, 2016.

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. *arXiv preprint arXiv:1902.07379*, 2019.

Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

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

Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. How does early stopping help generalization against label noise? *arXiv preprint arXiv:1911.08059*, 2019b.

Sainbayar Sukhbaatar, Joan Bruna, Manohar Paluri, Lubomir Bourdev, and Rob Fergus. Training convolutional networks with noisy labels. *arXiv preprint arXiv:1406.2080*, 2014.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1199–1208, 2018.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1701–1708, 2014.

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

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

Kiran Koshy Thekumparampil, Ashish Khetan, Zinan Lin, and Sewoong Oh. Robustness of conditional gans to noisy labels. *arXiv preprint arXiv:1811.03205*, 2018.

Sunil Thulasidasan, Tanmoy Bhattacharya, Jeff Bilmes, Gopinath Chennupati, and Jamal Mohd-Yusof. Combating label noise in deep learning using abstention. *arXiv preprint arXiv:1905.10964*, 2019.

Brendan Van Rooyen, Aditya Krishna Menon, and Robert C Williamson. Learning with symmetric label noise: The importance of being unhinged. *arXiv preprint arXiv:1505.07634*, 2015.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Andreas Veit, Neil Alldrin, Gal Chechik, Ivan Krasin, Abhinav Gupta, and Serge Belongie. Learning from noisy large-scale datasets with minimal supervision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 839–847, 2017.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016.

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

Yixin Wang, Alp Kucukelbir, and David M Blei. Robust probabilistic modeling with bayesian data reweighting. In *International Conference on Machine Learning*, pp. 3646–3655. PMLR, 2017.

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*, pp. 4524–4533, 2020a.

Zifeng Wang, Hong Zhu, Zhenhua Dong, Xiuqiang He, and Shao-Lun Huang. Less is better: Unweighted data subsampling via influence function. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6340–6347, 2020b.

Pengxiang Wu, Songzhu Zheng, Mayank Goswami, Dimitris Metaxas, and Chao Chen. A topological filter for learning with label noise. *arXiv preprint arXiv:2012.04835*, 2020a.

Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. Learning to purify noisy labels via meta soft label corrector. *arXiv preprint arXiv:2008.00627*, 2020b.

Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *arXiv preprint arXiv:1906.00189*, 2019.

Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33, 2020.

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

Forest Yang and Sanmi Koyejo. On the consistency of top-k surrogate losses. In *International Conference on Machine Learning*, pp. 10727–10735. PMLR, 2020.

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*, volume 33, pp. 9103–9110, 2019.

Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. *arXiv preprint arXiv:2006.07805*, 2020.

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

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

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Xuchao Zhang, Xian Wu, Fanglan Chen, Liang Zhao, and Chang-Tien Lu. Self-paced robust learning for leveraging clean labels in noisy data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6853–6860, 2020.

Zhilu Zhang and Mert R Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. *arXiv preprint arXiv:1805.07836*, 2018.

Guoqing Zheng, Ahmed Hassan Awadallah, and Susan Dumais. Meta label correction for noisy label learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, 2021.

Songzhu Zheng, Pengxiang Wu, Aman Goswami, Mayank Goswami, Dimitris Metaxas, and Chao Chen. Error-bounded correction of noisy labels. In *International Conference on Machine Learning*, pp. 11447–11457. PMLR, 2020.

A Theoretical Analysis

In this section, we provide a theoretical analysis of our proposed LT2L. First, we formally establish the need for label correction (Appendix A.1). Then, we provide a formal background for a statistically consistent classifier (Appendix A.2) and show the detailed calculation on the estimation of transition matrix $\boldsymbol{T}$ through forward propagation (Appendix A.3). Finally, we prove that $\boldsymbol{T}$ estimation error of our method are upper bounded (Appendix A.4).

A.1 Motivation: Need for Label Correction

Reducing the noise level of a dataset is crucial in learning with noisy labels both empirically (Drory et al., 2018; Zheng et al., 2021; Wu et al., 2020b; Hendrycks et al., 2018) and theoretically (Chen et al., 2019; Charikar et al., 2017). Following Chen et al. (2019); Ren et al. (2018); Jiang et al. (2018); Ma et al. (2018); Han et al. (2018b), the true transition matrix $\boldsymbol{T}$ in *symmetric noise* with noise level $\gamma\,(< 1)$ is defined as follows:

$$T_{ij} = \begin{cases} 1 - \left({}^{(N-1)}/_{N}\right)\gamma, & \text{for } i = j. \\ {}^{\gamma}/_{N}, & \text{otherwise.} \end{cases} \tag{8}$$

Correspondingly, the true transition matrix $\boldsymbol{T}$ in *asymmetric noise* with noise level $\gamma\,(< {}^{1}/_{2})$ is defined as follows:

$$T_{ij} = \begin{cases} 1 - \gamma, & \text{for } i = j. \\ \gamma, & \text{for some } i \neq j. \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

We employ these noise schemes in our CIFAR-10 experiments. Under the assumption that the label class is balanced, Chen et al. (2019) prove that the upper bound of test accuracy for the symmetric and asymmetric noise is as follows:

$$\begin{cases} \left({}^{(N-1)}/_{N}\right)\gamma^2 - 2\left({}^{(N-1)}/_{N}\right)\gamma + 1, & \text{for } \textit{symmetric noise}. \\ 2\gamma^2 - 2\gamma + 1, & \text{for } \textit{asymmetric noise}. \end{cases} \tag{10}$$

Eq. 10 shows quadratic (convex) functions of γ. In the case of the symmetric noise, test accuracy is minimized at the $\gamma = 1$. Similarly, with asymmetric noise, test accuracy is minimized at the $\gamma = {}^{1}/_{2}$. Hence, the test accuracy always decreases as γ increases in the feasible bound of γ. Therefore, reducing the noise level of a dataset plays a vital role in increasing the achievable test accuracy.

How to Reduce the Noise Level of a Dataset Two approaches are commonly used to reduce the noise level of a dataset: re-weighting samples and label correction. Sample re-weighting reduces the noise level by eliminating noisy samples during the model training, whereas label correction directly cleans up the dataset. Recently, label correction methods have shown notable results compared to sample re-weighting methods. Song et al. (2019a); Wu et al. (2020b); Zheng et al. (2021); Mirzasoleiman et al. (2020); Chang et al. (2017); Lin et al. (2017); Shrivastava et al. (2016) claim that re-weighting might show sub-optimal performance by filtering out noisy samples, which might aid in training feature extractors.

Theoretical Inspired Explanation of the Superiority of Label Correction Here, we provide a more theoretically motivated explanation of the above claim. We explore the reason behind the superior performance of label correction compared to sample re-weighting. Chen et al. (2019) considers only the upper bound of test accuracy according to the noise level while ignoring the effect on the number of samples on a generalization error while Charikar et al. (2017) does not consider deep networks. We aim to exhibit the superiority of label correction by presenting the generalization error considering both the number of samples and the noise level. We argue that both the noise level and the number of training samples are critical in determining the generalization error.

For simplicity, our explanation assumes binary classification with asymmetric noise with a level γ. We employ the VC dimension framework (Vapnik, 1999; 2013; Chen et al., 2020; Zhang et al., 2016) to describe the various methods for learning with noisy labels, although the framework provides a loose bound. Further investigation on a tighter bound using the Rademacher complexity or considering the multi-class classification is suggested for future research. Under clean training data distribution $\mathcal{D}$ and clean true data distribution $\mathcal{D}^*$, the VC dimension framework presents the following bound.

$$p\left(|\mathcal{E}_{\mathcal{D}}(f) - \mathcal{E}_{\mathcal{D}^*}(f)| > \epsilon\right) \leq 4\left(2|\mathcal{D}|\right)^{\mathbf{d}_{VC}} \exp\left(-\frac{1}{8}\epsilon^2|\mathcal{D}|\right), \tag{11}$$

where $\mathbf{d}_{VC}$ is the VC dimension and $\mathcal{E}_{\mathcal{D}}(f)$ is the expectation of error for function f regarding the data distribution $\mathcal{D}$. If the VC dimension $\mathbf{d}_{VC}$ is bounded (or finite), convergence is guaranteed because the upper bound decreases exponentially as the size of the dataset increases. Now, we observe a noisy dataset $\bar{\mathcal{D}}$ rather than a clean dataset $\mathcal{D}$. With the triangular inequality and the definition of γ, the following inequalities hold.

$$\begin{aligned}
&p\left(|\mathcal{E}_{\bar{\mathcal{D}}}(f) - \mathcal{E}_{\mathcal{D}^*}(f)| > \epsilon\right) && (12)\\
&= p\left(|\mathcal{E}_{\bar{\mathcal{D}}}(f) - \mathcal{E}_{\mathcal{D}}(f) + \mathcal{E}_{\mathcal{D}}(f) - \mathcal{E}_{\mathcal{D}^*}(f)| > \epsilon\right) && (13)\\
&\leq p\left(|\mathcal{E}_{\bar{\mathcal{D}}}(f) - \mathcal{E}_{\mathcal{D}}(f)| > \epsilon\right) + p\left(|\mathcal{E}_{\mathcal{D}}(f) - \mathcal{E}_{\mathcal{D}^*}(f)| > \epsilon\right) && (14)\\
&\leq \gamma + 4\left(2|\mathcal{D}|\right)^{\mathbf{d}_{VC}} \exp\left(-\frac{1}{8}\epsilon^2|\mathcal{D}|\right) && (15)
\end{aligned}$$

Even if this theoretical bound is loose, we argue that label correction shows better performance than re-weighting samples. As the dataset gets noisier, the number of filtered out samples by the re-weighting methods will also increase, resulting in a drastic reduction of the number of training samples. However, as aforementioned in Section 1, label correction holds an inherent problem of error propagation. We now explain the theoretical background on how the transition matrix acts as a safeguard for the label correction on our LT2L.

A.2 Background: Statistically Consistent Classification

It is well known that the label transition matrix T can be used to train *statistically consistent classifiers* in the presence of noisy labels (Mirzasoleiman et al., 2020; Xia et al., 2019; Yao et al., 2020). A statistically consistent classifier is a classifier which guarantees the convergence to an optimal classifier when the number of data samples increases indefinitely. Following Han et al. (2020a); Xia et al. (2019); Yao et al. (2020); Mirzasoleiman et al. (2020), we describe the consistency of empirical risk to yield the consistency of the classifier.

Statistically Consistent Empirical Risk Multi-class classification aims to train the hypothesis $\mathcal{H}$, which estimates a label y given an input x. Given the deep neural network $f_{\phi,\theta}$, a hypothesis $\mathcal{H}$ is commonly defined as follows:

$$\mathcal{H}(x) = \arg\max_{n\in\{1,\ldots,N\}} f_{\phi,\theta}(x)|_n. \tag{16}$$

With the true sample distribution $\mathcal{D}^*$, the *expected risk* $\mathcal{R}$ for $\mathcal{H}$ is defined as follows:

$$\mathcal{R}(\mathcal{H}) = \min_{\mathcal{H}} \mathbb{E}_{(x,y)\sim\mathcal{D}^*}\left[\mathcal{L}(\mathcal{H}(x), y)\right] \tag{17}$$

Under the distribution $\mathcal{D}^*$ is unknown, the optimal hypothesis $\mathcal{H}$ should minimize $\mathcal{R}$. Since the risk of the optimal hypothesis is difficult to calculate, the empirical risk is usually used for approximation via training dataset $\mathcal{D}$. The definition of empirical risk is as follows:

$$\mathcal{R}_{|\mathcal{D}|}(\mathcal{H}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathcal{L}(\mathcal{H}(x), y)\right] = \frac{1}{|\mathcal{D}|}\sum_{(x,y)\in\mathcal{D}} \mathcal{L}(\mathcal{H}(x), y). \tag{18}$$

Following equation holds for the *statistically consistent empirical risk*:

$$\mathcal{R}(\mathcal{H}) = \lim_{|\mathcal{D}|\to\infty} \mathcal{R}_{|\mathcal{D}|}(\mathcal{H}), \tag{19}$$

where it is common to assume that $\mathcal{D}$ is sampled from $\mathcal{D}^*$ as independent and identically distributed (i.i.d) random variables (Xia et al., 2019; Chen et al., 2019; 2020; Han et al., 2020a; Cheng et al., 2020).

Statistically Consistent Classifier Suppose an ideal zero-one loss function $\mathcal{L}^*$ (where it cannot be used in reality because its differentiation is impossible) (Bartlett et al., 2006):

$$\mathcal{L}^*(\mathcal{H}(x) = y) = \mathbf{1}_{\{\mathcal{H}(x) \neq y\}}. \tag{20}$$

$\mathbf{1}_{\{.\}}$ is an indicator function that outputs 1 if $\mathcal{H}(x) \neq y$ and 0 otherwise. If the class of the hypothesis $\mathcal{H}$ is large enough (Mohri et al., 2018), the optimal hypothesis to minimize the expected risk $\mathcal{R}(\mathcal{H})$ corresponds to the Bayes classifier (Bartlett et al., 2006) as follows:

$$\mathcal{H}(x) = \arg\max_{n \in \{1,\ldots,N\}} p(y = n|x) \tag{21}$$

Many classification loss functions in modern machine learning are proven to be *classification-calibrated* (Bartlett et al., 2006; Scott et al., 2012), i.e., the classification-calibrated loss function leads to a similar prediction to that of $\mathcal{L}^*$ when $|\mathcal{D}|$ is sufficiently large (Mohri et al., 2018; Vapnik, 2013). For example, the hinge loss is proven to be classification calibrated (Yang & Koyejo, 2020), and the cross-entropy loss with softmax function is empirically classification-calibrated (Guo et al., 2017). The classifier $f_{\phi,\theta}(x)$ is said to be statistically consistent when the classifier converges to the probability $p(y|x)$ by minimizing the empirical risk $\mathcal{R}_{|\mathcal{D}|}(\mathcal{H})$. Note that being risk consistent makes classifier consistent, but not vice versa (Xia et al., 2019).

Statistically Consistent Classifier in Noisy Labels The empirical risk $\mathcal{R}_{|\bar{\mathcal{D}}|}(\mathcal{H})$ of a noisy dataset $\bar{\mathcal{D}}$ is as follows.

$$\mathcal{R}_{|\bar{\mathcal{D}}|}(\mathcal{H}) = \mathbb{E}_{(x,y)\sim\bar{\mathcal{D}}}\left[\mathcal{L}(\mathcal{H}(x), y)\right] = \frac{1}{|\bar{\mathcal{D}}|}\sum_{(x,y)\in\bar{\mathcal{D}}} \mathcal{L}(\mathcal{H}(x), y) \tag{22}$$

Since the statistically consistent classifier $f_{\phi,\theta}(x)$ converges to $p(y|x)$, we can accept $f_{\phi,\theta}(x)$ to approximate $p(y|x)$. Given the definition of the transition matrix $p(\bar{y}|x) = T^\top p(y|x)$, a hypothesis with a noisy dataset $\bar{\mathcal{H}}$ is defined as follows:

$$\bar{\mathcal{H}}(x) = \arg\max_{n \in \{1,\ldots,N\}} T^\top f_{\phi,\theta}(x)|_n \tag{23}$$

Hence, minimizing the following empirical risk $\mathcal{R}_{|\bar{\mathcal{D}}|}(\bar{\mathcal{H}})$ using only the noisy dataset $\bar{\mathcal{D}}$ leads to a consistent classifier $f_{\phi,\theta}(x)$ (Xia et al., 2019).

$$\mathcal{R}_{|\bar{\mathcal{D}}|}(\bar{\mathcal{H}}) = \mathbb{E}_{(x,y)\sim\bar{\mathcal{D}}}\left[\mathcal{L}(\bar{\mathcal{H}}(x), y)\right] = \frac{1}{|\bar{\mathcal{D}}|}\sum_{(x,y)\in\bar{\mathcal{D}}} \mathcal{L}(\bar{\mathcal{H}}(x), y) \tag{24}$$

In other words, $f_{\phi,\theta}$ converges to the optimal classifier for the clean data when the sample size of the noisy dataset becomes infinitely large. Although other lines of research guarantee that maximizing accuracy in noisy data distribution maximizes accuracy in clean data distribution even without the transition matrix (Chen et al., 2020), loss correction via the transition matrix is still an effective consistent classifier training scheme. For this reason, a line of work in learning with noisy labels via the transition matrix attempts to train a statistically consistent classifier by an additional layer modeling the transition matrix preceded by the softmax layer (Goldberger & Ben-Reuven, 2016; Patrini et al., 2017; Thekumparampil et al., 2018; Yu et al., 2018; Mnih & Hinton, 2012; Reed et al., 2014; Sukhbaatar et al., 2014). Incidentally, it is known that modifying the loss function using the transition matrix has a degree of handling instance-dependent label corruption (Menon et al., 2016; Hendrycks et al., 2018).

Statistically Consistent Classifier in Noisy Labels with Small Clean Dataset We exploit a small number of clean data as in Finn et al. (2017); Veit et al. (2017); Lee et al. (2018); Jiang et al. (2018); Ren et al. (2018); Li et al. (2019); Hendrycks et al. (2018); Shu et al. (2019); Bahri et al. (2020); Zhang et al. (2020); Zheng et al. (2021); Wu et al. (2020b); Wang et al. (2020a) while disjointing the clean $\mathcal{D}$ and noisy dataset $\bar{\mathcal{D}}$. It is trivial that a statistically consistent classifier in exploiting a clean set can be obtained by minimizing the following empirical risk:

$$\mathcal{R}_{|\mathcal{D}|}(\mathcal{H}) + \mathcal{R}_{|\bar{\mathcal{D}}|}(\bar{\mathcal{H}}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\mathcal{L}(\mathcal{H}(x), y)\right] + \mathbb{E}_{(x,y)\sim\bar{\mathcal{D}}}\left[\mathcal{L}(\bar{\mathcal{H}}(x), y)\right] \tag{25}$$

$$= \frac{1}{|\mathcal{D}|}\sum_{(x,y)\in\mathcal{D}} \mathcal{L}(\mathcal{H}(x), y) + \frac{1}{|\bar{\mathcal{D}}|}\sum_{(x,y)\in\bar{\mathcal{D}}} \mathcal{L}(\bar{\mathcal{H}}(x), y). \tag{26}$$

Since the cross-entropy loss surrogates the ideal zero-one loss function $\mathcal{L}^*$ (Guo et al., 2017), minimizing the empirical risk $\mathcal{R}_{|\mathcal{D}|+|\bar{\mathcal{D}}|}(\mathcal{H}, \bar{\mathcal{H}})$ is equivalent to following optimization problem.

$$\arg\min_{\phi,\theta} \sum_{(x,y)\in\mathcal{D}} \mathcal{L}\left(f_{\phi,\theta}(x), y\right) + \sum_{(x,\bar{y})\in\bar{\mathcal{D}}} \mathcal{L}\left(\widehat{\boldsymbol{T}}^\top f_{\phi,\theta}(x), \bar{y}\right). \tag{27}$$

Without loss of generalization, the optimization problem can be rewritten by introducing an episodic batch formation in Section 3.1:

$$\arg\min_{\phi,\theta} \sum_{(x,y)\in d} \mathcal{L}\left(f_{\phi,\theta}(x), y\right) + \sum_{(x,\bar{y})\in\bar{d}} \mathcal{L}\left(\widehat{\boldsymbol{T}}^\top f_{\phi,\theta}(x), \bar{y}\right). \tag{28}$$

A.3 Calculation of the Estimated Transition Matrix $\widehat{T}$ of LT2L

GLC (Hendrycks et al., 2018) presents a method to estimate the transition matrix through a small clean dataset similar to our LT2L. GLC adopts the slow calculation method via a `FOR` or `WHILE` loop since Hendrycks et al. (2018) only requires to obtain the transition matrix once in the entire training process. However, our LT2L needs to estimate the transition matrix for every iteration as we correct the labels on the fly, ending up altering the ideal transition matrix. We speed up the estimation with a single forward propagation by using only matrix operations, avoiding the sluggish `FOR` or `WHILE` loop. Here, we show the derivation of Eq. 2. Let $d_i = \{(x, y) \in d | y_i = 1\}$. Then,

$$\widehat{T}_{ij} = p(\bar{y} = j | y = i) = \frac{1}{|d_i|} \sum_{(x,y)\in d_i} p(\bar{y} = j | y = i, x) = \frac{1}{|d_i|} \sum_{(x,y)\in d_i} f_{\phi,\theta}(x) \Bigg|_j \tag{29}$$

$$= \frac{1}{|d_i|} \left(\sum_{(x,y)\in d_i} y f_{\bar{\phi},\bar{\theta}}(x)^\top \Bigg|_{(i,j)} \right) = \left(\sum_{(x,y)\in d_i} y f_{\bar{\phi},\bar{\theta}}(x)^\top \Bigg|_{(i,j)} \right) \frac{1}{|d_i|} \tag{30}$$

$$= \left(\sum_{(x,y)\in d_i} y f_{\bar{\phi},\bar{\theta}}(x)^\top \Bigg|_{(i,j)} \right) \left(\mathrm{diag}^{-1} \left(\sum_{(x,y)\in d} y \right) \Bigg|_{(i,j)} \right) \tag{31}$$

Without loss of generality, $\widehat{\boldsymbol{T}}$ can be written as follows:

$$\widehat{\boldsymbol{T}} = \left(\sum_{(x,y)\in d} y f_{\bar{\phi},\bar{\theta}}(x)^\top \right) \mathrm{diag}^{-1} \left(\sum_{(x,y)\in d} y \right). \tag{32}$$

A.4 Proof of Theorem 1

In this section, we prove under strong assumptions (Theorem 2) followed by milder assumptions (Theorem 1). Theorem 2 estimates the upper bound of the error on transition matrix T, assuming the ideal situation where $p(\bar{y}|x)$ is perfectly parameterized to $f_{\bar{\phi},\bar{\theta}}(x)$.

Theorem 2. *Assuming $p(\bar{y}|x) = f_{\bar{\phi},\bar{\theta}}(x)$, for $\epsilon \geq 0$,*

$$p\left(\left|\widehat{T}_{ij} - T_{ij}\right| > \epsilon\right) \leq 2\exp\left(-2\epsilon^2 K\right). \tag{33}$$

Proof. If $p(\bar{y}|x) = f_{\bar{\phi},\bar{\theta}}(x)$, then $p\left(\left|\mathbb{E}\left[\widehat{\boldsymbol{T}}\right] - \boldsymbol{T}\right| > \epsilon\right) = 0$. With the triangular and Hoeffding inequality, the following holds:

$$p\left(\left|\widehat{T}_{ij} - T_{ij}\right| > \epsilon\right) = p\left(\left|\widehat{T}_{ij} - \mathbb{E}\left[\widehat{T}_{ij}\right] + \mathbb{E}\left[\widehat{T}_{ij}\right] - T_{ij}\right| > \epsilon\right) \tag{34}$$

$$\leq p\left(\left|\mathbb{E}\left[\widehat{T}_{ij}\right] - T_{ij}\right| > \epsilon\right) + p\left(\left|\widehat{T}_{ij} - \mathbb{E}\left[\widehat{T}_{ij}\right]\right| > \epsilon\right) \tag{35}$$

$$= p\left(\left|\mathbb{E}\left[\widehat{T}_{ij}\right] - T_{ij}\right| > \epsilon\right) + 2\exp\left(-2\epsilon^2 K\right) \tag{36}$$

$$= 2\exp\left(-2\epsilon^2 K\right). \tag{37}$$

□

With Theorem 2 alone, we can see that the estimation error of transition matrix $\boldsymbol{T}$ decreases exponentially as K (the number of samples per class) increases, as we mentioned in Theorem 1 of Section 3.2.

We assume the hypothetical case that $p(\bar{y}|x)$ could flawlessly model $f_{\bar{\phi},\bar{\theta}}(x)$, but the assumption does not hold in practice. Several lemmas are established in order to prove Theorem 1 under more relaxed assumptions. If $p(\bar{y}|x) \neq f_{\bar{\phi},\bar{\theta}}(x)$, then $p(|\mathbb{E}[\widehat{\boldsymbol{T}}] - \boldsymbol{T}| > \epsilon) \neq 0$. We focus on examining the upper bound of $p(|\mathbb{E}[\widehat{\boldsymbol{T}}] - \boldsymbol{T}| > \epsilon)$ under the relaxed assumption. The upper bound of $\widehat{\boldsymbol{T}}$ (which is equivalent to $f_{\bar{\phi},\bar{\theta}}(x)$) is strictly 1 since it is a probability. By applying McDiarmid's concentration inequality (Boucheron et al., 2013), the following inequality is established:

$$p\left(\left|\mathbb{E}\left[\widehat{T}_{ij}\right] - T_{ij}\right| > \epsilon\right) \leq \mathbb{E}_\sigma\left[\sup_{\mathcal{H}} \frac{1}{|\bar{\mathcal{D}}|} \sum_{(x,\bar{y})\in\bar{\mathcal{D}}} \sigma_x \mathcal{L}(\mathcal{H}(x), \bar{y})\right] + \sqrt{\frac{\log(1/\epsilon)}{2|\bar{\mathcal{D}}|}} \tag{38}$$

where σ is an i.i.d Rademacher random variable (Montgomery-Smith, 1990) and $\mathcal{H}$ is a hypothesis.

We estimate the upper bound of the estimation error of $\boldsymbol{T}$ by assuming $\mathcal{H}$ is constructed using deep neural networks. A deep neural networks hypothesis $\mathcal{H}'$ is defined as follows.

$$\mathcal{H}'(x) = \bar{\theta}\mathcal{A}_{H-1}(\bar{\phi}_{H-1}\mathcal{A}_{H-2}(\dots \mathcal{A}_1(\bar{\phi}_1 x))) \in \mathbb{R}^N \tag{39}$$

where H is the depth of deep neural networks and $\mathcal{A}_i$ is the i-th activation function. When the function class is limited with deep neural networks, the following lemma holds by borrowing the results of (Xia et al., 2019).

Lemma 1. *Suppose σ is an i.i.d Rademacher random variable and $\mathcal{L}$ is the cross-entropy loss function which is L-Lipschitz continuous with respect to $\mathcal{H}'$,*

$$\mathbb{E}_\sigma\left[\sup_{\mathcal{H}} \frac{1}{|\bar{\mathcal{D}}|} \sum_{(x,\bar{y})\in\bar{\mathcal{D}}} \sigma_x \mathcal{L}(\mathcal{H}(x), \bar{y})\right] \leq NL\mathbb{E}_\sigma\left[\sup_{\mathcal{H}'} \frac{1}{|\bar{\mathcal{D}}|} \sum_{(x,\bar{y})\in\bar{\mathcal{D}}} \sigma_x \mathcal{H}'(x)\right] \tag{40}$$

where $\mathcal{H}'$ is a hypothesis belonging to the function class of deep neural networks.

As opposed to using the VC dimension framework in Section 1, this section uses the Rademacher complexity framework (Bartlett & Mendelson, 2002) to assess the upper bounds of our method. Hypothesis complexity of deep neural networks via Rademacher complexity is broadly studied in Xia et al. (2019); Bartlett et al. (2017); Golowich et al. (2018); Neyshabur et al. (2017). In particular, Golowich et al. (2018) proves the following lemma:

Lemma 2. *Assume the Frobenius norm of the weight matrices $\bar{\phi}_1, ..., \bar{\phi}_{H-1}, \bar{\theta}$ are at most $\bar{\Phi}_1, ..., \bar{\Phi}_{H-1}, \bar{\Theta}$ for H-layer neural networks $f_{\bar{\phi},\bar{\theta}}$. Let the activation functions be 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Let σ is an i.i.d Rademacher random variable. Let x is upper bounded by B, i.e., for any $x \in \mathcal{X}$, $\|x\| \leq B$. Then, for $\epsilon \geq 0$*

$$\mathbb{E}_\sigma\left[\sup_{\mathcal{H}'} \frac{1}{|\bar{\mathcal{D}}|} \sum_{(x,\bar{y})\in\bar{\mathcal{D}}} \sigma_x \mathcal{H}'(x)\right] \leq \frac{B(\sqrt{2H\log 2} + 1)\bar{\Theta}\Pi_{h=1}^{H-1}\bar{\Phi}_i}{\sqrt{|\bar{\mathcal{D}}|}}. \tag{41}$$

Now, we can complete the proof of Theorem 1.

Theorem 1. *Assume the Frobenius norm of the weight matrices $\bar{\phi}_1, ..., \bar{\phi}_{H-1}, \bar{\theta}$ are at most $\bar{\Phi}_1, ..., \bar{\Phi}_{H-1}, \bar{\Theta}$ for H-layer neural networks $f_{\bar{\phi},\bar{\theta}}$. Let the loss function be L-Lipschitz continuous w.r.t. $f_{\bar{\phi},\bar{\theta}}$. Let the activation functions be 1-Lipschitz, positive-homogeneous, and applied element-wise (such as the ReLU). Let x is upper bounded by B, i.e., for any $x \in \mathcal{X}$, $\|x\| \leq B$. Then, for $\epsilon \geq 0$*

$$p\left(\left|\widehat{T}_{ij} - T_{ij}\right| > \epsilon\right) \leq \frac{NLB(\sqrt{2H\log 2} + 1)\bar{\Theta}\Pi_{h=1}^{H-1}\bar{\Phi}_i + \sqrt{1/2\log(1/\epsilon)}}{\sqrt{|\bar{\mathcal{D}}|}} + 2\exp\left(-2\epsilon^2 K\right).$$

Proof. With the triangular inequality, Hoeffding inequality, Theorem 2, Lemma 1, and 2, the following holds.

$$p\left(\left|\widehat{T}_{ij} - T_{ij}\right| > \epsilon\right) \tag{42}$$

$$= p\left(\left|\widehat{T}_{ij} - \mathbb{E}\left[\widehat{T}_{ij}\right] + \mathbb{E}\left[\widehat{T}_{ij}\right] - T_{ij}\right| > \epsilon\right) \tag{43}$$

$$\leq p\left(\left|\mathbb{E}\left[\widehat{T}_{ij}\right] - T_{ij}\right| > \epsilon\right) + p\left(\left|\widehat{T}_{ij} - \mathbb{E}\left[\widehat{T}_{ij}\right]\right| > \epsilon\right) \tag{44}$$

$$= p\left(\left|\mathbb{E}\left[\widehat{T}_{ij}\right] - T_{ij}\right| > \epsilon\right) + 2\exp\left(-2\epsilon^2 K\right) \tag{45}$$

$$\leq \mathbb{E}_\sigma\left[\sup_{\mathcal{H}} \frac{1}{|\bar{\mathcal{D}}|} \sum_{(x,\bar{y})\in\bar{\mathcal{D}}} \sigma_x \mathcal{L}(\mathcal{H}(x), \bar{y})\right] + \sqrt{\frac{\log(1/\epsilon)}{2|\bar{\mathcal{D}}|}} + 2\exp\left(-2\epsilon^2 K\right) \tag{46}$$

$$\leq NL\mathbb{E}_\sigma\left[\sup_{\mathcal{H}'} \frac{1}{|\bar{\mathcal{D}}|} \sum_{(x,\bar{y})\in\bar{\mathcal{D}}} \sigma_x \mathcal{H}'(x)\right] + \sqrt{\frac{\log(1/\epsilon)}{2|\bar{\mathcal{D}}|}} + 2\exp\left(-2\epsilon^2 K\right) \tag{47}$$

$$\leq \frac{NLB(\sqrt{2H\log 2}+1)\bar{\Theta}\Pi_{h=1}^{H-1}\bar{\Phi}_i}{\sqrt{|\bar{\mathcal{D}}|}} + \sqrt{\frac{\log(1/\epsilon)}{2|\bar{\mathcal{D}}|}} + 2\exp\left(-2\epsilon^2 K\right) \tag{48}$$

$$\leq \frac{NLB(\sqrt{2H\log 2}+1)\bar{\Theta}\Pi_{h=1}^{H-1}\bar{\Phi}_i + \sqrt{1/2\log(1/\epsilon)}}{\sqrt{|\bar{\mathcal{D}}|}} + 2\exp\left(-2\epsilon^2 K\right). \tag{49}$$

□

Theorem 1 and 2 state that the estimation error of transition matrix $\boldsymbol{T}$ is reduced with a larger K. However, we experimentally verify that $K = 1$ is enough for achieving comparable performance (See Appendix C.6.2).

We end this section by enumerating the limitations of our theoretical analysis. (i) Although our method is based on a multi-head architecture, a clean and a noisy classifier are trained simultaneously, where only the training of the noisy classifier is considered in the theoretical analysis. (ii) We failed to make the upper bound tight for Theorem 1 and 2. Additional assumptions like Charikar et al. (2017) may yield tighter bound. We conjecture that our method works empirically well for $K = 1$ since the upper bound is loose. (iii) The data distribution on a noisy classifier changes in every iteration due to simultaneously corrected labels. However, we assume that the data distribution is stationary. In order to make our theoretical assumption more adequate for our method, it is necessary to examine the situation under changing data distribution.

B Experimental Details

B.1 Datasets

As shown in Table 4, we use bigger noisy dataset (noisy-train) and smaller clean dataset (clean-train) for training. Validation set is used for obtaining the best model. Since CIFAR datasets do not have the validation set, we split 10% of the entire training set as the validation set. Thus, experimental results may differ from the results of their papers. For Clothing1M, we use its original data split.

Table 4: Data split composition of dataset used in our experiments.

Dataset	Noisy-train	Clean-train	Valid	Test	Image size	# of classes
CIFAR-10 CIFAR-100	44K	1K	5K	10K	32 × 32	10 100
Clothing1M	1M	47K	14K	10K	224 × 224	14

B.2 MINI-BATCH CONSTRUCTION

We sample the mini-batches from both clean and noisy datasets. For the clean dataset, we construct the mini-batch to have the same number of instances per class for clean samplers, whereas the mini-batch of the noisy set is randomly sampled. We choose the batch size 100 for both CIFAR-10 and CIFAR-100, a total of 200 images used per iteration. As the number of classes is 10 and 100, 10 and 1 image(s) are used per class for the clean batch, respectively. We choose the batch size 42 for each noisy and clean set on Clothing1M dataset with 14 classes so that 84 images are used per iteration, where 3 samples are used per class for the clean batch.

B.3 DETAILED TRAINING PROCEDURE

B.3.1 CIFAR-10/100 DATASET

Here, we describe the detailed training procedure of our baselines on CIFAR-10/100 dataset (Krizhevsky et al., 2009). For all baselines except Deep kNN, we use SGD optimizer with an initial learning rate of 1e-1. For Deep kNN (Bahri et al., 2020), we use Adam optimizer (Kingma & Ba, 2014) and set the initial learning rate to 1e-3. We follow the experimental settings described in each corresponding papers as much as possible to obtain performance fairly.

L2RW: When training the model, we decay the learning rate to 1e-2 and 1e-3 at 40 and 60 epochs, with a total of 80 epochs. **MW-Net:** For the total of 60 epochs, we decay the learning rate by a factor of 10 at 40 and 50 epochs, respectively. **Deep kNN:** Since it has multiple training stages, we first train two independent models with only the clean dataset $\mathcal{D}$ and sum of the clean and noisy dataset $\mathcal{D} \cup \bar{\mathcal{D}}$, respectively. Then, we filter out the suspicious samples from the noisy set to generate the filtered noisy set $\bar{\mathcal{D}}_{\text{filter}}$ using k-nearest neighbors algorithm (k-NN) of the logit outputs from one of the two trained models, where we choose the model with the better validation set accuracy. Finally, we train the model with the sum of the clean and filtered noisy set $\mathcal{D} \cup \bar{\mathcal{D}}_{\text{filter}}$. For each phase, we train the model until 100 epochs without learning rate decay. **GLC:** We first train the model with only the noisy set $\bar{\mathcal{D}}$ and obtain the label transition matrix with the trained model. With the label transition matrix, we train the initialized model with the clean and noisy set $\mathcal{D} \cup \bar{\mathcal{D}}$ while correcting the loss obtained from the noisy samples. **MLoC:** We first train the model with a warm-up of 75 epochs, i.e., directly training with the noisy label dataset without bells and whistles. We then meta-train the model with the learning rate of 1e-4 for additional 75 epochs. **MLaC:** For a total of 120 epochs, we decay the learning rate at 80 and 100 epochs by a factor of 10. **MSLC:** Similar to MLoC, we first train the model with the warm-up of 80 epochs, then meta-train the model with the learning rate of 1e-2 and cut it to 1e-3 at 20 epochs, for 40 epochs. **LT2L (Ours):** we train the model until 70 epochs and decay the learning rate at 50 and 60 epochs by a factor of 10.

B.3.2 CLOTHING1M DATASET

For the Clothing1M dataset (Xiao et al., 2015), we borrow the baseline evaluation results from each corresponding paper except for L2RW, where we train the model ourselves as the original paper does not report the results. For a fair comparison, we use the same backbone network, pre-trained ResNet-50. **L2RW:** We train the model for 10 epochs using the SGD optimizer with the initial learning rate of 1e-2. We decay the learning rate after 5 epochs by a factor of 10, where we follow the common training procedure borrowed from (Wu et al., 2020b; Shu et al., 2019). **LT2L (Ours):** Similarly, we use the SGD optimizer with the same initial learning rate, where we decay the learning rate after 1 epochs for total of 2 epochs.

B.4 EVALUATION DETAILS FOR SECTION 4.3

Considering the situation where we have to purify the existing noisy labels inside the training set automatically, predicting the correct labels of the train samples is crucial. We compare the accuracy on the noisy train dataset where we compare with the clean label, which is unknown to the model at training time. We show the accuracy on CIFAR-10 with symmetric noise of 80%; hence if the model is perfectly overfitted to the noisy set, it will yield 28% accuracy.

Settings. For each method, we use the model with the best validation accuracy, i.e., the best model that each has produced. For the meta-model of MLaC, we use the output of the label correction

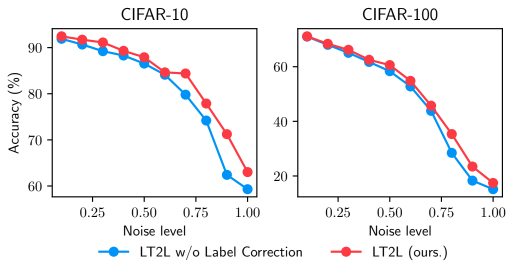

Figure 4: Effect of label correction on CIFAR-10/100 with various symmetric noise ratio. Accuracy (%) of LT2L (ours.) and LT2L without label correction is provided.

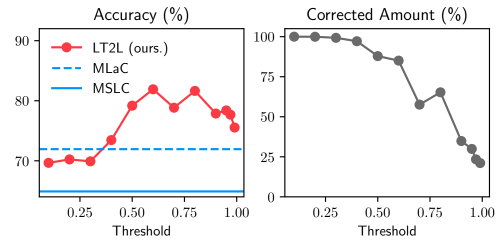

Figure 5: Robustness to Mis-labeling of LT2L. The corrected label amount (%) and model accuracy (%) according to the label correction threshold (ρ) are provided.

network (LCN), where the network is fed with the feature vector and the noisy label for every noisy dataset to yield a corrected soft label. The feature vector is extracted from the main model, where it is obtained using the features before the fully connected layer. For the meta-model of MSLC, we use the cached soft label from the last epoch. MSLC calculates the soft label with the linear combination of the previous cached soft label, the predicted label from the main model, and the given noisy label, where the weights are continuously learned during training.

Results. We analyze the performance in terms of two factors: accuracy, which is the direct performance of the model, and NLL, which is the target of the noisy label training. We check using all the samples (Overall) and only the wrongly labeled samples (Incorrect). As we described earlier, our method shows superior performance in both accuracy and NLL compared to all the other methods that we have evaluated. We can also observe the efficacy of our label correction method. Furthermore, the performance of the MSLC meta-model is near 28%, implying the incorrectness of the meta-model.

B.5 Baselines in Figure 5

Figure 5 demonstrates the robustness of our method when unreliable samples are also corrected by lowering the threshold of label correction to investigate how safe our method is. Our method uses the transition matrix to avoid the error propagation problem even if unreliable samples are corrected. We observe the robustness of our method compared to other label correction methods, MLaC (Zheng et al., 2021) and MSLC (Wu et al., 2020b).

C Additional Experiments

C.1 Robustness to Miscorrected Labels: What Happens if Labels are Wrongly Corrected?

This section illustrates the robustness of our label correction method to miscorrected labels by comparing it with other label correction methods (MLaC and MSLC) in learning with noisy labels. We examine whether miscorrection accumulates errors to deteriorate the predictive performance. To simulate the label correction error, we perturb the corrected labels by injecting artificial noise. We control the degree of random perturbation to observe the robustness of each method on various levels of failure.

Settings. We experiment on CIFAR-10 with symmetric 80% noise where the label correction would be the most critical on performance. We further test the robustness for our LT2L and MSLC by comparing it with the performance obtained without label correction.

Results. Figure 6 reveals the susceptibility of MLaC on miscorrection by showing steep performance degradation when perturbation worsens. MSLC shows trivial performance gains when labels are corrected, implying that it is not using the full benefits of label correction. Furthermore, when highly perturbed, MSLC performance worsens if it attempts to correct the labels. In contrast, the

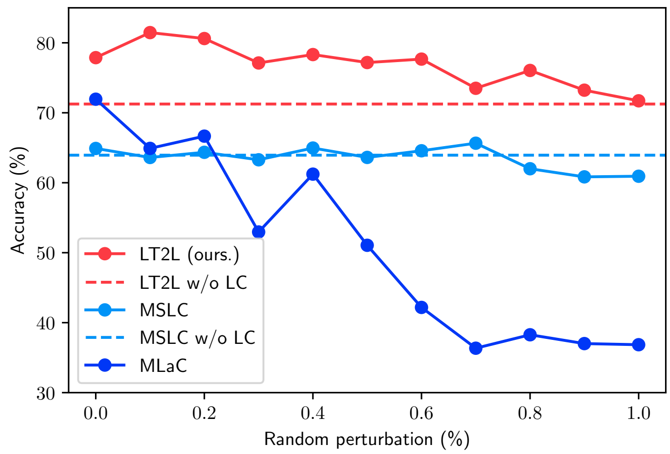

Figure 6: Robustness to miscorrected labels on CIFAR-10 with various perturbation strength. Test accuracy (%) of baselines and baselines without the label correction is provided.

label correction of our LT2L shows stable performance improvement even in harsh situations. We can understand this result as our method being highly robust to potentially wrong corrections.

C.2 Training Time Comparison

With the Figure 1, the exact values in Table 5 further affirm the efficiency of our method, substantially outperforming the baseline. Table 5 shows seconds per single iteration and the total training hours of each baseline, including our LT2L. Since Deep kNN and GLC require multiple training stages, the sum of the iteration times of each training phase is provided.

Table 5: Training time comparison on CIFAR-10 dataset with 80% symmetric noise. Time (seconds) per iteration and Time (hours) per total training on a single RTX 2080Ti GPU are provided with the relative ratio compared to our method.

Method	L2RW	MW-Net	Deep kNN	GLC	MLoC	MLaC	MSLC	**LT2L (ours.)**
Iteration Time (Relative to Ours.)	0.416 (5.84x)	0.298 (4.19x)	0.127 (1.78x)	0.0846 (1.19x)	0.422 (5.92x)	0.527 (7.39x)	0.279 (3.91x)	**0.0713**
Total Training Time (Relative to Ours.)	4.78 (3.11x)	2.63 (1.71x)	2.32 (1.51x)	2.29 (1.49x)	7.13 (4.64x)	10.2 (6.64x)	2.51 (1.64x)	**1.54**

C.3 Is the Performance Improvement Due to Over-sampling on the Clean Dataset?

Unlike many MAML-based methods using the clean dataset as gradient guidance in the meta training step, our proposed method utilizes the dataset directly during the model training. One may suspect that the performance improvement of our method may come from over-sampling the clean dataset. Therefore, we compare our proposed model with an over-sampling method (Chawla et al., 2002; Hong et al., 2020). To see the effectiveness of our batch formation, we experiment with the standard cross-entropy loss (Eq. 50) instead of our final objective (Eq. 6), using the same batch formation (Naïve Oversampling). For a fair comparison, label correction is excluded.

$$\arg\min_{\phi,\theta} \sum_{(x,y)\in d} \mathcal{L}\left(f_{\phi,\theta}(x), y\right) + \sum_{(x,\bar{y})\in \bar{d}} \mathcal{L}\left(f_{\phi,\theta}(x), \bar{y}\right) \tag{50}$$

Table 6 shows that our proposed method outperforms the over-sampling method. This observation indicates that our meta-learning method appropriately leverages the clean dataset to estimate the label corruption matrix.

Table 6: The effect of clean set oversampling on the performance of CIFAR-10/100 experiments. The accuracy (%) of naïve oversampling and LT2L (ours.) w/o label correction is provided.

Dataset	Method	Symmetric				Asymmetric	
		20 %	40 %	60 %	80 %	20 %	40 %
CIFAR-10	Naïve Oversampling	89.39	85.90	83.90	56.83	90.58	84.41
	LT2L (ours.) w/o Label Correction	**90.67**	**88.29**	**84.12**	**74.19**	**92.50**	**91.05**
CIFAR-100	Naïve Oversampling	65.42	57.08	42.18	25.88	67.71	61.73
	LT2L (ours.) w/o Label Correction	**68.02**	**61.75**	**52.79**	**28.46**	**69.59**	**66.07**

C.4 Comparison to Other Methods with the Transition Matrix

Although the transition matrix is initially introduced as a safeguard to mitigate the risk of label correction in the LT2L, our LT2L even shows better performance than other methods employing the transition matrix. This section illustrates that LT2L, even without label correction, shows better performance than other methods using transition matrix with the clean dataset: GLC (Section C.4.1) and MLoC (Section C.4.2).

C.4.1 Comparison to Gold Loss Correction (GLC) (Hendrycks et al., 2018)

Our proposed method is similar to GLC in estimating the label transition matrix, but it shows better performance than GLC even without label correction (See Table 7). Additionally, instead of estimating the transition matrix, we directly use the oracle matrix to examine the effectiveness of the multi-head architecture more clearly. Even using the same oracle matrix for both methods, our LT2L outperforms GLC. We conjecture that our multi-head architecture trains the model to extract features better than the two-stage training of GLC, which learns noisy classifier and clean classifier consecutively.

Table 7: The effect of two-head architecture via oracle label transition matrix on CIFAR-10/100 dataset. Test accuracy (%) of GLC Hendrycks et al. (2018) and LT2L (ours.) with and without oracle label transition matrix is provided. For a fair comparison, label corruption is excluded in LT2L.

Dataset	Method	Symmetric				Asymmetric	
		20 %	40 %	60 %	80 %	20 %	40 %
CIFAR-10	GLC Hendrycks et al. (2018) w/ Oracle	89.06	85.45	81.56	67.54	91.74	90.35
	LT2L (ours.) w/ Oracle w/o Label Correction	**91.37**	**88.71**	**83.97**	**74.91**	**91.80**	**91.10**
	GLC Hendrycks et al. (2018)	89.66	85.30	80.34	67.44	91.56	89.76
	LT2L (ours.) w/o Label Correction	**90.67**	**88.29**	**84.12**	**74.19**	**92.50**	**91.05**
CIFAR-100	GLC Hendrycks et al. (2018)	60.99	49.00	33.38	20.38	64.43	54.20
	LT2L (ours.) w/o Label Correction	**68.02**	**61.75**	**52.79**	**28.46**	**69.59**	**66.07**

C.4.2 Comparison to Meta Loss Correction (MLoC) (Wang et al., 2020a)

Since our method does not directly parameterize the label transition matrix $\boldsymbol{T}$, stable estimation of $\boldsymbol{T}$ and its theoretical analysis are possible (See Theorem 1). Table 8 shows that MLoC and our LT2L without Label Correction (LC) shows comparable performance. MLoC uses several engineering techniques for stable training: a strong prior and gradient clipping, where it is not mentioned in the paper. However, our method shows good performance even without label correction, being robust to different hyperparameters, reducing the need for excessive engineering. We also emphasize that there is a significant gap in performance at a severe noise level.

C.4.3 Empirical Convergence Analysis in T Estimation

This section analyzes the convergence of estimation error between the oracle transition matrix $\boldsymbol{T}$ and the estimated transition matrix $\widehat{\boldsymbol{T}}$, comparing our LT2L to other methods, MLoC and GLC,

Table 8: Test accuracy (%) comparison between LT2L without Label Correction and MLoC Wang et al. (2020a) on CIFAR-10/100 dataset.

Dataset	Method	Symmetric 20 %	Symmetric 40 %	Symmetric 60 %	Symmetric 80 %	Asymmetric 20 %	Asymmetric 40 %
CIFAR-10	MLoC Wang et al. (2020a)	90.50	87.20	81.95	54.64	91.15	89.35
	LT2L (ours.) w/o Label Correction	**91.37**	**88.71**	**83.97**	**74.91**	**91.80**	**91.10**
CIFAR-100	MLoC Wang et al. (2020a)	**68.16**	**62.09**	**54.49**	20.23	69.20	**66.48**
	LT2L (ours.) w/o Label Correction	68.02	61.75	52.79	**28.46**	**69.59**	66.07

which learn the transition matrix. Label correction is excluded for our LT2L as it may produce unfair comparisons. Figure 7 shows the difference between the probability distribution of the oracle transition matrix and the estimated transition matrix for each iteration, where Pearson χ^2-divergence is used to measure the discrepancy between the two matrices. Since GLC estimates the transition matrix only once in the entire learning process, it represents a fixed value unrelated to iteration. The plot for the MLoC error seems to be constant, but it is, in fact, slowly decreasing. It implies that MLoC is likely to be highly dependent on the initialization of the transition matrix $\widehat{\boldsymbol{T}}$. Although our LT2L does not require pre-training and uses only clean samples inside a single mini-batch, it shows fast convergence with a similar estimation error to GLC.

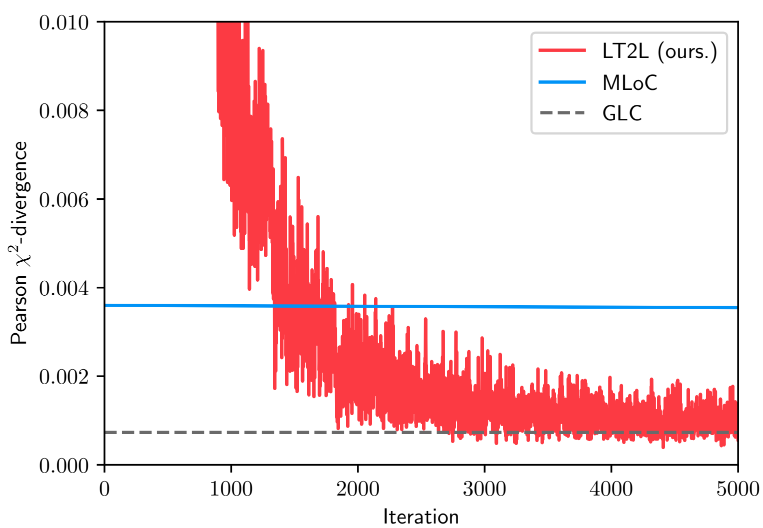

Figure 7: Plot of transition matrix estimation error for every iteration. Pearson χ^2-divergence of our LT2L, MLoC and GLC is provided.

C.5 Incorrect Label Detection Performance Comparison

We consider the case where we have to continuously purify the already-collected dataset with the existence of a human oracle, where the process can be accelerated by correctly detecting the candidates for the wrongly labeled samples. Hence, we regard the incorrect label detection problem as a binary classification problem where the model output probability of the noisy samples is used as the barometer for the correctness of the label.

Settings. We extract the probability values of the noisy labels per sample inside the noisy training set to be the negative score for the binary classification problem where we label 1 for the wrongly labeled sample and 0 otherwise. We additionally measure the performance of the meta-models. We use the meta-learned sample weights for MSLC, MW-Net, and L2RW. For MLaC, we use the probability of the soft label obtained by the meta-model, which is described in detail in Appendix B.4. Finally, as Deep kNN filters out the doubtful samples while training the final model, we regard the process as weighting each sample by 0 or 1 depending on its doubtfulness. Note that we evaluate each method using all the training samples, including the correctly labeled, as each method may mistake those samples to be wrongly labeled.

Results. Ren et al. (2018); Shu et al. (2019); Bahri et al. (2020) claim that using meta-learning or pre-training is able to tell whether a sample is mislabeled. Although our model is not directly aimed at finding noisy samples in the noisy set, Table 9 shows that our model achieves comparable or better performance in detecting noisy labels than the baselines. Although Ren et al. (2018); Bahri et al. (2020) claim that the performance has improved because the meta-model detects noisy samples through re-weighting, the actual performance of meta-models is generally lower than that of the final classifier. It implies that Ren et al. (2018); Bahri et al. (2020) may operate with different dynamics than the original author intended.

Table 9: Incorrect label detection performance comparison on CIFAR-10 with symmetric 80% noise. The Area Under the Receiver Operating Characteristic (AUROC) and The Area Under the Precision-Recall Curve (AUPRC) are provided. Note that pure random model will yield 0.5 AUROC and 0.72 AUPRC. † denotes the performance of the sample weights obtained with the meta model.

	L2RW	L2RW†	MW-NET	Deep kNN	Deep kNN†	GLC	MLoC	MLaC	MLaC†	MSLC	**LT2L**
AUROC	0.8653	0.4898	0.9205	0.9019	0.8070	0.9324	0.9318	0.9640	0.9564	0.9303	**0.9651**
AUPRC	0.9412	0.7994	0.9631	0.9396	0.9326	0.9674	0.9695	**0.9835**	0.9791	0.9624	**0.9835**

C.6 Analysis on Our Method

C.6.1 How Many Clean Samples are Required?

To verify the effect of the clean dataset size, we observe the performance differences while varying the size. As shown in Table 10, our method consistently shows better performance on different sizes. Especially, even when our method only uses 100 clean samples, it outperforms all the baselines which utilize all the clean samples (1,000 samples). This observation demonstrates that our method could accurately estimate the label transition matrix with a small number of clean samples, which can be applicable to real-world scenarios where it is difficult to obtain a sufficient number of clean samples.

C.6.2 How Sensitive is to the Batch Size?

In Section 3.2, we show that the accuracy of estimating the label transition matrix is upper-bounded by the number of samples in the mini-batch. As previous studies (Hendrycks et al., 2018) mentioned, the quality of the estimated transition matrix affects the performance in learning with noisy labels. To verify the effect of the number of samples in the mini-batch, we observe the performance changes by varying the number of samples per class in the mini-batch from 1 to 10. As shown in Table 11, there is little change in performance depending on the number of samples per class, although the performance degradation is predicted by Theorem 1 when the number of samples is small. From this observation, we believe that our proposed method shows practicality even in situations where the batch size cannot be increased due to the limited computing resources.

C.6.3 Searching the Optimal Hyperparameter λ

We observe performance variance on the CIFAR-10/-100 datasets when we change the hyperparameter λ which is a loss balancing factor. The results are summarized in Table 12. The hyperparameter is searched in $\{0.01, 0.05, 0.1, 0.2, 0.5, 1.0\}$.

D Additional Related Work

D.1 Comparison with Other Methods with Label Transition Matrix

Under the assumption that label corruption occurs class-dependently and instance-independently, learning with noisy label methods exploiting the label transition matrix has shown admirable performance (Mnih & Hinton, 2012; Reed et al., 2014; Sukhbaatar et al., 2014; Bekker & Goldberger, 2016;

Table 10: The effect of the number of clean set on CIFAR-10 with symmetric 80% noise. Comparison with other label correction methods with meta-learning is provided.

# of clean examples	MLaC	MSLC	**LT2L (ours.)**
100	32.92	69.00	**76.48**
250	42.15	63.52	**79.36**
500	50.70	63.35	**77.82**
1000	71.94	64.90	**77.88**

Table 11: The effect of the number of samples per class (K) in the mini-batch on the predictive performance (Accuracy (%)) of CIFAR-10 experiments.

K	Symmetric				Asymmetric	
	20 %	40 %	60 %	80 %	20 %	40 %
2	91.85	89.96	87.00	81.68	92.43	91.11
4	91.79	89.85	86.92	82.18	92.14	91.18
6	92.16	90.07	86.77	79.57	92.14	90.98
8	92.10	89.82	84.81	79.55	92.41	90.74
10	91.72	89.30	84.63	77.88	91.95	90.25

Table 12: Evaluation results varying the hyperparameter λ. Test accuracy (%) with 95% confidence interval of 5-runs is provided.

	λ	Symmetric Noise Level				Asymmetric Noise Level	
		20%	40%	60%	80%	20%	40%
CIFAR-10	0.01	90.87 ± 0.35	89.03 ± 0.24	85.20 ± 1.00	75.95 ± 1.13	91.06 ± 0.36	89.40 ± 0.44
	0.05	91.69 ± 0.17	89.20 ± 0.64	84.48 ± 0.89	76.39 ± 1.79	91.92 ± 0.31	89.58 ± 0.31
	0.1	91.72 ± 0.20	89.30 ± 0.32	84.63 ± 0.70	77.88 ± 1.09	91.95 ± 0.22	90.25 ± 0.39
	0.2	91.72 ± 0.11	89.61 ± 0.29	85.71 ± 0.24	77.55 ± 2.78	92.20 ± 0.19	90.51 ± 0.27
	0.5	91.94 ± 0.28	90.07 ± 0.17	86.78 ± 0.31	79.52 ± 0.78	92.29 ± 0.10	90.43 ± 0.31
	1.0	91.80 ± 0.20	89.70 ± 0.19	86.66 ± 0.48	80.95 ± 0.44	92.04 ± 0.40	90.54 ± 0.23
CIFAR-100	0.01	65.36 ± 0.77	57.79 ± 1.12	43.65 ± 1.06	26.95 ± 0.76	66.58 ± 0.71	62.19 ± 0.66
	0.05	68.49 ± 0.27	62.47 ± 0.32	53.55 ± 0.86	35.53 ± 1.28	69.73 ± 0.18	65.63 ± 0.79
	0.1	68.38 ± 0.29	62.53 ± 0.33	54.82 ± 0.46	35.35 ± 1.13	69.35 ± 0.13	66.34 ± 0.27
	0.2	68.65 ± 0.09	63.07 ± 0.22	54.84 ± 0.30	35.65 ± 0.66	70.37 ± 0.15	66.93 ± 0.20
	0.5	68.75 ± 0.60	63.82 ± 0.33	55.22 ± 0.64	37.36 ± 1.15	70.35 ± 0.51	67.93 ± 0.53
	1.0	67.91 ± 0.59	62.78 ± 0.28	52.76 ± 1.15	31.45 ± 0.75	70.02 ± 0.60	67.11 ± 0.55

Patrini et al., 2017; Goldberger & Ben-Reuven, 2016). It is well known that training a statistically consistent classifier is possible if the transition matrix is estimated accurately, but precise estimation is usually challenging (Mirzasoleiman et al., 2020; Xia et al., 2019; Yao et al., 2020). Various methods have been proposed to alleviate the issue: imposing strong prior (Patrini et al., 2017; Han et al., 2018a), designing a loss function using the ratio of the label transition matrix $\boldsymbol{T}$ (Xia et al., 2019), or factorization of the transition matrix (Yao et al., 2020). However, it is still challenging to estimate the transition matrix with only the noisy dataset. Recently, approaches that improve the estimation accuracy of the transition matrix using a small clean dataset have shown remarkable results: Gold Loss Correction (GLC) (Hendrycks et al., 2018) and Meta Loss Correction (MLoC) (Wang et al., 2020a). The clean dataset makes it possible to directly estimate the noisy label posterior, resulting in stable prediction of the transition matrix. Our LT2L differs from the existing methods which try to find the fixed label transition matrix, as the oracle transition matrix continuously changes during label correction in our method. Our method shows novelty compared to the previous two methods even without the label correction.

Differences from Gold Loss Correction (GLC) (Hendrycks et al., 2018) Similar to LT2L, GLC models $p(\bar{y}|x)$ as $f_{\bar{\phi},\bar{\theta}}(x)$ for estimating the transition matrix $\boldsymbol{T}$. However, GLC is more inefficient than our LT2L because it requires multiple training phases (See § 4.2 and Appendix C.2). We introduce a multi-head architecture with to speed up the training. Furthermore, there is an additional performance advantage compared to GLC. The multi-head architecture is presumed to help obtain a better feature extractor by inducing corruption-independent feature extraction. Detailed experimental results can be found in Appendix C.4.1.

Differences from Meta Loss Correction (MLoC) Wang et al. (2020a) MLoC gradually finds the oracle transition matrix $\boldsymbol{T}$ via the MAML framework (Finn et al., 2017). As mentioned earlier, MLoC is very slow because it requires three back-propagations for a single iteration due to its nature of MAML (See § 4.2 and Appendix C.2). MLoC directly parameterizes the transition matrix T and learns it using various engineering techniques: strong prior and gradient clipping, which were not mentioned in original paper. In contrast, our method estimates $\boldsymbol{T}$ more accurately by sampling the posterior through a single forward propagation. We empirically validate that our method performs better or comparable to MLoC even without label correction (See Appendix C.4.2).

D.2 Methods using Multi-head Architecture for Noisy Labels

We propose a multi-head architecture to estimate the transition matrix efficiently: one is for the clean label distribution, and the other is for the noisy label distribution. A similar multi-head architecture has been used in situations dealing with crowdsourcing. Many crowdsourcing studies assume that multiple people label a single image (Rodrigues & Pereira, 2018; Guan et al., 2018; Tanno et al., 2019), where training a reliable classifier is the goal of the crowdsourcing problem. They maintain separate heads for each annotator, and each head performs multi-task learning to learn each annotator's decisions directly. Then, the final decision is made by voting each head's decision. There is no component for estimating the label transition matrix in these methods and no primary head classifier to learn from the estimated label transition matrix.

D.3 Meta Learning without Multiple Back-propagation in a Single Iteration

Meta-learning is roughly divided into optimization- (Finn et al., 2017; Grant et al., 2018; Lee & Choi, 2018) and metric-based methods (Snell et al., 2017; Sung et al., 2018; Vinyals et al., 2016). Although the optimization-based methods led by MAML (Finn et al., 2017) are now the mainstream, they require a tremendous computational cost with multiple back-propagations for a single iteration, making them hard to use in practice, where metric-based methods do not have this problem. However, metric-based methods are highly optimized only for few-shot classification tasks.

Note that LT2L can hardly be regarded as their variants, since there is no inner-optimization loop or a k-NN classifier. To devise an efficient meta-learning algorithm, we borrow the idea of weight sharing from Vinyals et al. (2016) ($\phi = \bar{\phi}$) but use different parameters for θ and $\bar{\theta}$ in order to compute the transition matrix. While most of the methods using meta-learning in learning with noisy labels have focused on the model agnostic advantages of MAML, we propose a novel meta-learning method for learning with noisy labels. We define the task as estimating the transition matrix with small subset of the clean dataset. Thus, meta-module $\bar{\phi}$ in our framework is episodically optimized through these tasks.