EFFICIENTLY META-LEARNING FOR ROBUST DEEP NETWORKS WITHOUT PRIOR UNBIASED SET

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning with noisy labels is a practically challenging problem in robust deep learning. Recent efforts to improve the robustness are made by meta-learning the sample weights or transition matrix from a *prior unbiased set*. Thus, previous meta-learning based approaches generally assume the existence of such prior unbiased set. Unfortunately, this assumption unrealistically simplifies the task of learning noisy labels in real-world scenarios; even worse the updating iterations in previous meta-learning algorithms typically demand prohibitive computational cost. This paper proposes an efficient meta-learning approach for robust deep learning to address these challenges. Specifically, without relying on prior unbiased validation set, our method dynamically estimates unbiased samples in training data and leverages meta-learning to refine the deep networks. Furthermore, to significantly reduce the updating iterations in optimization cost, we elaborately design the inner loop adaption and outer loop optimization of the meta-learning paradigm, respectively. Experimental results demonstrate that our approach is able to save about 6 times training time while achieving comparable or even better generalization performance. In particular, we improve accuracy on the CIFAR100 benchmark at 40% instance-dependent noise by more than 13% in absolute accuracy.

1 INTRODUCTION

Training set biases are inevitable in real-world scenarios, and in particular, noisy labels can negatively influence the model performance. Critically, recent studies (Zhang et al., 2017; Arpit et al., 2017; Toneva et al., 2019) have shown that deep neural networks (DNNs) can overfit to label noise and eventually lead to the poor generalization performance. Thus, as an effective learning paradigm, meta-learning (Finn et al., 2017; Nichol et al., 2018) has been widely studied for noisy label learning (Ren et al., 2018). The underlying concept of 'learning to learn' is to train the model at the meta-level beyond to achieve the data-agnostic and noise-type-agnostic meta-models. This facilitates the model adapting to the specific tasks.

Recently, there is a growing interest in meta-learning-based methods to optimize sample weights (Ren et al., 2018), model parameters (Li et al., 2019), or noise transition matrix Q (Wang et al., 2020), and corrected labels (Zheng et al., 2021). However, the applicability of these meta-learning-based methods in real-world scenarios is hindered by two limitations. First, as shown in Figure 1(left), the vanilla meta-learned robust model demands a *prior unbiased* subset to find the optimal sample-weight or heuristically approximates Q. Unfortunately, the assumption of an additional unbiased subset available is pretty strong in practice, thus limiting the general applicability and deployment of meta-learned models. Therefore, it is essential in this paper to study how to directly estimate unbiased data from the biased training data. Second, existing meta-learning methods (Ren et al., 2018; Shu et al., 2019; Zhang et al., 2020; Zheng et al., 2021) require the nested optimization of models and other parameters, which takes the expensive second-order computation (Finn et al., 2017). In particular, we highlight two computational bottlenecks: (1) dot product between training and validation inputs, and (2) dot product between training and validation gradient directions. This computational burdens prevent these methods from using large-scale DNNs. We thus emphasize on further improving the efficiency of the model optimization.

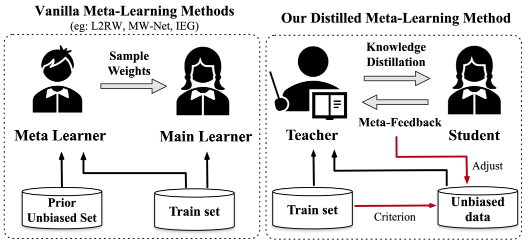

Figure 1: Overview of our proposed method compared with the vanilla meta-learning methods.

To this end, we propose an efficiently meta-learning approach for robust deep networks that does not only rely on prior unbiased subset, but also significantly reduce the optimization cost as well. Particularly, as shown in Figure 1(right), the key idea is to leverage meta-feedback mechanism to correct the sample bias by observing how its inaccurate-labels would affect the student. We propose meta-learning to dynamically distill unbiased knowledge based on the learning state of the model in a teacher-student manner. This is motivated by the recent success – DINO (Caron et al., 2021) of knowledge distillation with no labels. Specifically, we introduce a novel meta-feedback mechanism which adjusts the distillation criterion of biased samples by minimizing the loss of the student network on distilled samples. The proposed mechanism consists of two nested stages, where the inner stage, *i.e*., the student network, consists of one-step gradient descent for knowledge distillation, and the outer stage updates teacher parameters based on student feedback.

Our approach naturally fits robust deep learning and can handle the aforementioned two challenges. For the *first* challenge, our method can dynamically distill unbiased samples in training data and does not rely on a prior unbiased set, making the new method more practical. We show that using the unbiased samples which are dynamically selected based on the learning state of the student will result in better performance than using a prior unbiased set. For the *second* challenge, the huge optimization cost of second-order meta gradient (Nichol et al., 2018), we introduce a distillation objective in the inner loop adaption to approximate multi-step adaptation and propose an approximate solution to the second-order gradient in the outer loop optimization. Section 5.2 indicates that our proposed method can save about 6 times of training time. Meanwhile, Section 5.1 shows that the proposed method is able to outperform or achieve highly competitive performance compared with other gradient-based methods with noisy labels, especially on the CIFAR100 benchmark of 40% instance-dependent noise with more than 13% absolute accuracy. Additionally, we further extend and verify the effectiveness of our proposed method in the setting of class imbalance. Surprisingly, our new method can also achieve very competitive results compared with other gradient-based methods.

In summary, the contribution of this paper is three-fold:

1) We present an efficiently meta-learning approach, which eliminates the dependence on additional unbiased data and reduces the optimization complexity of recent meta-learning based method.
2) A novel meta-feedback mechanism is presented, which can dynamically update the teacher network and refine estimators based on the learning state of the student network. In order to reduce the computational complexity, we refine the inner loop adaption and outer loop optimization of the meta-learning paradigm respectively.
3) Experimental results demonstrate that our approach is able to save about 6 times the training time while still achieving a comparable or even better generalization performance in both label noise and long-tail settings. Furthermore, we analyze the insights of our approach in the risk of distillation object and theoretical guarantees for the convergence of the proposed approach.

2 Related work

Learning with noisy labels has observed exponentially growing interests. These methods address this problem from a variety of perspectives. For example, several works focus on estimating the noise transition matrix Q to correct the predictions (Menon et al., 2015; Goldberger & Ben-Reuven, 2017; Patrini et al., 2017; Hendrycks et al., 2018; Xia et al., 2019). Since the CrossEntropy loss has been proved to easily overfit noisy labels (Zhang et al., 2017), Ghosh et al. (2017); Zhang & Sabuncu (2018); Wang et al. (2019); Ma et al. (2020) try to design different loss functions to reduce the influence of noisy samples or regularization terms (Zhang et al., 2018; Hu et al., 2020; Menon et al., 2020; Liu et al., 2020). Another direction seeks to improve the label quality by correcting the noisy labels (Lee et al., 2018; Tanaka et al., 2018; Yi & Wu, 2019). Moreover, some methods choose the clean data via the 'small-loss' trick (Han et al., 2018; Jiang et al., 2018; Yu et al., 2019; Wei et al., 2020), but this introduces selection criterion hyperparameters. Liu & Tao (2015); Jiang et al. (2018; 2020); Fang et al. (2020); Zhang et al. (2021) proposes to reduce the weights assigned to noisy samples.

Among them, meta-learning (Finn et al., 2017; 2018; 2019; Zintgraf et al., 2019; Raghu et al., 2020) has recently emerged as an effective framework for robust deep learning. In addition, through an additional unbiased validation subset, meta-learning based methods (Ren et al., 2018; Shu et al., 2019; Chen et al., 2021; Zhang et al., 2020; Xu et al., 2021; Wu et al., 2021a) can cope distribution shift, including label noise and class imbalance. For example, L2RW (Ren et al., 2018) directly adjusts the weight for each example. MW-Net (Shu et al., 2019) learns an explicit weighting function. FaMUS (Xu et al., 2021) learns to approximate the meta gradient. MLC (Wang et al., 2020) estimates the noise transition matrix. These methods all learn a meta-model from a prior unbiased set but differ in specific ways to correct the biased training labels. Unfortunately, this paradigm unrealistically simplifies the task of noisy label learning in real-world scenarios. MLNT (Li et al., 2019) simulates regular training with synthetic noisy labels. Unlike the MLNT, which lacks a clear signals of clean supervision, we encourage the teacher to adjust the target distribution of training sample explicitly. Knowledge distillation (Hinton et al., 2015; Romero et al., 2015) contracts a softened softmax probability distribution to transfer the knowledge, which inspired their subsequent works (Li et al., 2017; Zhang et al., 2020) that leveraging a predefined unbiased set to optimize exemplar weights and labels of mislabeled samples in order to distill effective supervision. Instead of directly applying KD loss, inspired by the idea of knowledge distillation, we propose to integrate KD into the meta-learning paradigm, treating the unbiased data as knowledge to dynamically distill.

3 Methodology

Notations. Consider a classification problem with the training set $\mathcal{D} = \{(x_i, y_i)|_{i=1}^N\}$, where x_i denotes the i-th sample, $y_i \in \{0,1\}^c$ is the label vector over c classes, and N is the number of the entire training data. Let $\mathcal{X} \in \mathbb{R}^d$ be the feature space and $\mathcal{Y} = \{1 \dots c\}$ be the label space, where each $(x_i, y_i) \in (\mathcal{X} \times \mathcal{Y})$. A classifier is a function that maps input feature space to the label space $\mathcal{F} : \mathcal{X} \to \mathbb{R}^c$. Let $\mathcal{F}(\cdot \mid \theta_T)$ and $\mathcal{F}(\cdot \mid \theta_S)$ be the teacher network and the student network in our approach while θ_T and θ_S denote their parameters respectively. We use $\ell_{ce}(p, q)$ to denote the cross-entropy loss between two distributions q and p. And $\mathrm{KL}(\cdot, \cdot)$ denotes the Kullback-Leibler divergence.

Meta-Feedback Mechanism.

Our approach is summarized in Figure 2. It naturally fits the bi-level optimization framework, where the student network θ_S is the inner-learner and the teacher network θ_T is the meta-learner. In particular, the proposed meta-feedback mechanism consists of two nested stages, where the inner stage consists of one-step gradient descent for knowledge distillation, and the outer stage updates teacher parameters based on student feedback. This two-stage paradigm is named meta-feedback mechanism in this work. Specifically, we encourage the teacher to adjust the target distribution of training samples in a manner that minimizes the loss of the student network on distilled samples. The teacher network is updated by policy gradients computed by evaluating the student network on distilled samples. The complete algorithm is shown in Algorithm 1.

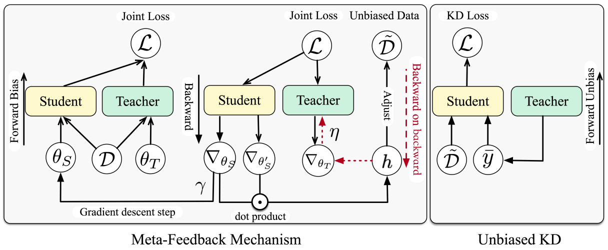

Figure 2: The flowchart of our Distilled Meta-Learning approach.

Meta Objective. We take the above meta-feedback mechanism as our meta-objective of optimization and formalize it as,

$$\begin{aligned} &\min_{\theta_T} \mathbb{E}_{(\tilde{x},\tilde{y}) \in \widetilde{\mathcal{D}}} \mathcal{L}\left(\tilde{y}, \mathcal{F}(\tilde{x} \mid \theta_S^*(\theta_T))\right), \\ &\text{s.t. } \theta_S^*(\theta_T) = \arg\min_{\theta_S} \mathbb{E}_{\bar{y} \sim P(\cdot \mid x; \theta_T)} \mathcal{L}\left(\bar{y}, \mathcal{F}(x \mid \theta_S)\right) \end{aligned} \tag{1}$$

where $\widetilde{\mathcal{D}} = \{(\tilde{x}_i, \tilde{y}_i)|_{i=1}^M\}$ indicates the unbiased samples distilled from teacher, and the distillation criterion is controlled by the feedback of student. And $\bar{y}$ indicates the target distribution of training samples from teacher network. Note that the meta-optimization is performed over the θ_T, whereas the objective is computed using the updated θ_S. In effect, our proposed method aims to optimize teacher parameters such that one gradient step on unbiased samples will produce maximally effective behavior on that task.

Specifically, our proposed loss function $\mathcal{L}$ on $\mathcal{D}$ is constructed as follows,

$$\mathcal{L}(\mathcal{D}; \theta_T; \theta_S) = (1 - \lambda) \cdot \mathcal{L}_{sup} + \lambda \cdot \mathcal{L}_{KD} \tag{2}$$

where λ is a hyper-parameter to control the relative importance of the two terms. In this loss function, the first part $\mathcal{L}_{sup} = \ell_{ce}(\mathcal{F}(x_i \mid \theta_S), y_i) + \ell_{ce}(\mathcal{F}(x_i \mid \theta_T), y_i)$ is supervised learning loss of the student and teacher networks on biased $\mathcal{D}$, the second part $\mathcal{L}_{KD} = \mathrm{KL}\left(p^{\xi}, q^{\xi}\right) = \mathrm{KL}\left(\mathcal{F}(x_i \mid \theta_S, \xi), \mathcal{F}(x_i \mid \theta_T, \xi)\right)$ refers to the Knowledge Distillation (Hinton et al., 2015) loss which makes the logits of student match the ones of teacher. DNNs typically produce class probabilities by using a softmax output layer that converts the logit, z_i, computed for i-th class into a probability, q_i, by comparing z_i with the other logits: $q_i = {}^{exp(z_i/\xi)}\!/\!_{\sum_j exp(z_j/\xi)}$, where ξ is the temperature factor to control the importance of each soft target. The optimization details are described below.

Meta Learning Optimization. Intuitively, by optimizing the teacher's parameters according to the performance of the student on unbiased samples, the target distribution can be adjusted accordingly to further improve student's performance. As we are effectively trying to optimize the teacher on a meta level, the optimal student's parameters should minimize loss on distilled samples. For each training step, we give a mini-batch of training samples x and the learning rate γ, and the student network is updated with knowledge distillation algorithms:

$$\theta_S'(\theta_T) = \theta_S - \gamma \nabla_{\theta_S} \mathcal{L}(x; \theta_S; \theta_T) \tag{3}$$

For example, when using one gradient update at t epoch, the update rules are as follows:

$$\theta_S^{(t+1)} = \theta_S^{(t)} - \gamma \nabla_{\theta_S} \ell_{ce}\left(\bar{y}, \mathcal{F}(x \mid \theta_S)\right)|_{\theta_S = \theta_S^{(t)}} \tag{4}$$

where $\bar{y} \sim P(\cdot \mid x; \theta_T)$ indicates pseudo labels distilled from the teacher. In this work, we sample the hard pseudo labels from the teacher distribution to train the student.

The meta-optimization across tasks is performed via Stochastic Gradient Descent (SGD), such that the θ_T is updated as follows:

$$\theta_T' = \theta_T - \eta \nabla_{\theta_T} \mathcal{L}\left(\theta_S - \nabla_{\theta_S} \mathcal{L}(\theta_S, \theta_T)\right) \tag{5}$$

This differentiation requires computing the gradient of gradient [1]. To simplify notation, we will consider one gradient update for the rest of this section, but using multiple gradient updates is a straightforward extension.

Practical Approximation. As mentioned above, the student uses the teacher's distillation knowledge to update its parameters. In expectation, the new parameter of student is $\mathbb{E}_{\bar{y} \sim P(\cdot \mid x; \theta_T)}\left[\theta_S - \gamma \nabla_{\theta_S} \mathcal{L}\left(\bar{y}, \mathcal{F}(x \mid \theta_S)\right)\right] := \bar{\theta}_S'$. We will update the teacher's parameter to minimize the student's loss on a batch of unbiased data. Then, by the chain rule:

$$\begin{aligned} \frac{\partial \mathcal{R}}{\partial \theta_T} &= \frac{\partial}{\partial \theta_T} \mathcal{L}\left(\mathcal{F}(\tilde{x} \mid \bar{\theta}_S'), \tilde{y}\right) \\ &= \left.\frac{\partial \mathcal{L}\left(\mathcal{F}(\tilde{x} \mid \bar{\theta}_S'), \tilde{y}\right)}{\partial \theta_S}\right|_{\theta_S = \bar{\theta}_S'} \cdot \frac{\partial \bar{\theta}_S'}{\partial \theta_T} \propto \gamma \cdot h \cdot \frac{\mathcal{L}\left(\bar{y}, \mathcal{F}(x; \theta_T)\right)}{\partial \theta_T} \end{aligned} \tag{6}$$

where $h = \frac{\partial \mathcal{L}(\tilde{y}, \mathcal{F}(\tilde{x}; \theta_S))}{\partial \theta_S} \cdot \left(\frac{\partial \mathcal{L}(\bar{y}, \mathcal{F}(x; \theta_S))}{\partial \theta_S}\right)^{\top}$ is the dot product of the gradient from unbiased training loss and the gradient from the knowledge distillation loss. Intuitively, h quantifies how much a specific change in θ_T affects the training gradient of the student. As a result, it is also known as the student's feedback. We refer readers to Appendix A for the detailed derivation.

Meta-Feedback. We rewrite the computation of the feedback of student network coefficient h:

$$\tilde{h} = \left[\nabla_{\theta_S'} \mathcal{L}\left(\tilde{y}, \mathcal{F}(\tilde{x}; \theta_S')\right)\right]^{\top} \cdot \nabla_{\theta_S} \mathcal{L}\left(\bar{y}, \mathcal{F}(x; \theta_S)\right) \tag{7}$$

[1] Since θ_T depends on θ_S via Equation (5), we compute its second-order dependency on θ_T.

In this work, $\tilde{h}$ is not only used to approximate the meta-gradient, but also dynamically affect the unbiased samples selection criteria. Intuitively, $\tilde{h}$ quantifies the reward signal of how well the student performs on distilled samples, indicating the estimation accuracy of unbiased data. Since we use the $\tilde{h}$ to compute selection criteria, we get rid of the prior assumption about the noise rate τ compared to (Malach & Shalev-Shwartz, 2017; Han et al., 2018; Yu et al., 2019; Wei et al., 2020).

Unbiased Knowledge Distillation. Before introducing the details, we first clarify the connection between small losses and clean instances. Intuitively, small-loss examples are likely to be the ones that are correctly labeled (Han et al., 2018; Jiang et al., 2018; Yu et al., 2019; Wei et al., 2020). Based on this, applying the 'small-loss' criterion can select 'clean' instances, but it cannot eliminate data bias. For example, in noise-robust training problems, this additional validation set $\mathcal{D}^{val}$ is expected to have clean and class-balanced labels (Shu et al., 2019). Although (Malach & Shalev-Shwartz, 2017; Han et al., 2018; Yu et al., 2019; Wei et al., 2020) propose sample-selection-based method and achieve promising performance, the assumptions in previous work cannot always hold true. In other words, $\mathcal{D}^{val}$ is required to be unbiased and to have a reasonable size. Following the setting of Han et al. (2018), previous work update $R(t)$ (the ratio of small-loss samples), which controls how many samples should be selected in each mini-batch. Therefore, we take that $R(t)$ should be data-driven and related to the learning state of the student network.

Algorithm 1 The Proposed Distilled Meta-Learning

```
Input: Train set $\mathcal{D} = \{(x_i, y_i)|_{i=1}^{N}\}$, teacher $\theta_T$, student $\theta_S$, teacher and student learning rate $\eta$, $\gamma$, epoch $T$.
Initialize $\theta_T^{(0)}$ and $\theta_S^{(0)}$.
for $t = 0$ to $T$ do
    Fetch mini-batch $\mathcal{D}_n$ from $\mathcal{D}$.
    Obtain $\widetilde{\mathcal{D}}$ by $R_t(c)|\mathcal{D}_n|$.
    Sample a pseudo label $\bar{y} \sim P(\cdot \mid x_i; \theta_T)$.
    One step gradient update:
    $\theta_S^{(t+1)} = \theta_S^{(t)} - \gamma \nabla_{\theta_S} \ell_{ce}\left(\bar{y}, \mathcal{F}\left(x; \theta_S\right)\right)|_{\theta_S = \theta_S^{(t)}}$
    Compute the student's feedback $h$ by Equation (7)
    Compute $g_T^{(t)} = h \cdot \nabla_{\theta'_T} \ell_{ce}\left(\hat{y}, \mathcal{F}\left(x; \theta_T\right)\right)|_{\theta_T = \theta_T^{(t)}}$
    update teacher: $\theta_T^{t+1} = \theta_T^{t} - \eta \cdot g_T^{(t)}$
    update $R_t\left(c\right)$ by Equation (8)
end for
return $\theta_S^{(T)}$
```

$$R_t\left(c\right) = \sum_{b=1}^{B}\left(\left[\!\!\left[\arg\min_{\mathcal{D}_n} \mathcal{L}\left(\mathcal{D}_n; \theta_T; \theta_S\right) > \sum_{t=1}^{T} \tilde{h}_t + \epsilon_0 \right]\!\!\right] \cdot [\![\arg\max(\mathcal{F}(x \mid \theta_T)) = c]\!] \right) \tag{8}$$

where $[\![]\!]$ is the Iverson's bracket notation and B indicates batch size. $\sigma_t(c)$ reflects the selection criterion of class c at time step t. $R_c(t) = \epsilon_0 + \sum_{i=0}^{c} \tilde{\sigma}_t(i)$. Here we formalize $R_c(t)$ as a selection criterion for c classes (see Figure 3(c) and Figure 4 for ablation studies). In order to select the sample at step $t = 0$, we use a very small ϵ as the initial. The ablation study for ϵ can be found in Figure 3(f).

Why estimate unbiased knowledge instead of prior unbiased set? In this work, we propose a learning-based approach that learns $R_t(c)$ by employing a meta-learning optimization strategy. Intuitively, the prior unbiased set reflects the finite field classes distribution, which is biased from the real distribution, while estimating the unbiased sample from the training sample provides a path to approximate the real distribution. It is worth noting that the teacher and student networks have different learning routes, so the outputs of the two models will not increasingly become consistent. To illustrate the core idea of our proposed meta-feedback mechanism, we formalize the algorithm as Algorithm 1, and we also provide the pseudo code of our proposed method, see details in Algorithm 2.

4 Theoretical Insights

In this section, we analyze the convergence and complexity of Algorithm 1. In particular, we need to handle the difference between two losses over the training and unbiased set (*i.e.*, inner- and outer-loop losses) in the analysis.

Convergence analysis. And then, we provide the convergence and complexity analysis for Algorithm 1 based on the properties established in the previous subsection.

Theorem 4.1. *Let Assumption D.1 and Assumption D.2 hold, and apply Algorithm 1 to solve the objective function Equation* (1). *Under the same setting of Theorem D.5, choose* $\gamma = \frac{1}{8L}, C_\eta = 80.$

Table 1: Test accuracy (%) on CIFAR10/CIFAR 100 with symmetric noise. M indicates the number of clean samples required for the method.

Method	CIFAR10				CIFAR100			
	M	$\tau = 0.2$	$\tau = 0.4$	$\tau = 0.6$	M	$\tau = 0.2$	$\tau = 0.4$	$\tau = 0.6$
Forward	×	87.90 ± 0.00	-	-	×	58.60 ± 0.00	-	-
GLC	1k	91.43 ± 0.00	88.52 ± 0.00	84.08 ± 0.00	1k	69.30 ± 0.00	63.24 ± 0.00	56.12 ± 0.00
GCE	×	89.90 ± 0.20	87.10 ± 0.20	-	×	66.80 ± 0.40	61.80 ± 0.20	-
PENCIL	×	-	-	-	×	73.86 ± 0.34	69.12 ± 0.62	57.79 ± 3.86
Co-teaching	×	82.32 ± 0.07	74.81 ± 0.00	73.06 ± 0.00	×	54.23 ± 0.08	46.20 ± 0.00	35.67 ± 0.00
JoCoR	×	85.73 ± 0.19	-	-	×	53.01 ± 0.04	-	-
DivideMix	×	96.10 ± 0.00	-	-	×	77.30 ± 0.00	-	-
MentorNet	1k	92.00 ± 0.00	89.00 ± 0.00	-	1k	73.00 ± 0.00	68.00 ± 0.00	-
MentorMix	1k	95.60 ± 0.00	94.20 ± 0.00	91.30 ± 0.00	1k	78.60 ± 0.00	71.30 ± 0.00	64.60 ± 0.00
L2RW	1k	90.00 ± 0.40	86.92 ± 0.19	82.24 ± 0.36	1k	67.10 ± 0.10	61.34 ± 2.06	48.15 ± 0.34
MW-Net	1k	-	89.27 ± 0.28	84.07 ± 0.33	1k	-	67.73 ± 0.26	58.75 ± 0.11
MLC	0.5k	84.28 ± 0.00	79.89 ± 0.00	-	0.5k	-	-	-
IEG	0.1k	96.20 ± 0.20	95.90 ± 0.20	-	1k	81.20 ± 0.70	80.20 ± 0.30	-
FaMUS	1k	-	95.37 ± 0.15	**94.97** ± **0.11**	1k	-	75.91 ± 0.19	73.58 ± 0.28
FastRW	×	95.10 ± 0.10	93.70 ± 0.10	-	×	78.70 ± 0.20	74.20 ± 0.40	-
Ours	×	**96.92** ± **0.32**	**95.94** ± 0.21	94.60 ± 0.20	×	**82.26** ± **0.23**	**80.76** ± **0.18**	**74.20** ± **0.43**

We have

$$\mathbb{E}\|\nabla\mathcal{L}(w_\zeta)\| \leq \mathcal{O}\left(\frac{1}{T} + \frac{\sigma^2}{B} + \sqrt{\frac{1}{T} + \frac{\sigma^2}{B}}\right) \tag{9}$$

Theorem 4.1 shows that the proposed method converges linearly with the number T of outer-loop meta iterations, and the convergence error decays linearly with the number B of sampled mini-batch. See proof in Appendix D. Our proposed distilled meta-learning converges sublinearly with the convergence error decaying sublinearly with the number of samples due to nonconvexity of the meta objective function. The empirical results also support our conclusion Figure 3(d).

Risk analysis. Next, we analyze the risk of students learning teacher-generated soft-targets in the distillation meta-learning framework. Similar to Li et al. (2017), we define the expected risk $\mathcal{R}_y$ associated with the unreliable label sampled from $\mathcal{D}$: $\mathcal{R}_{\hat{y}} = \mathbb{E}_{\mathcal{D}}\left[\ell(\hat{y}, y^*)\right]$. where y^* is the unknown ground-truth label, and $\mathbb{E}$ denotes the expectation over $\mathcal{D}$. Then, we show that the risk of using Equation (1) can be smaller than using either the full noisy labels or only the partial clean labels.

Theorem 4.2. *The optimal risk associated with $\bar{y}^\lambda$ is smaller than both risks with $\hat{y}$ and $\tilde{y}$, i.e.,*

$$\min_\lambda \mathcal{R}_{\bar{y}^\lambda} < \min\{\mathcal{R}_{\hat{y}}, \mathcal{R}_{\tilde{y}}\}, \textit{ for } \lambda = \frac{\mathcal{R}_{\tilde{y}}}{\mathcal{R}_{\hat{y}} + \mathcal{R}_{\tilde{y}}} \tag{10}$$

where $\hat{y}$ is the unreliable label, and $\tilde{y}$ is the clean label.

Theorem 4.2 indicates that, by properly setting the balance weight λ, we can obtain a pseudo label that is closer to the ground-truth label in theory. Therefore, we can potentially train a better model based on our distillation framework. See proof in Appendix C.

5 Experiments

We compare our method with other methods in the label noise setting in Section 5.1, including symmetric noise, asymmetric noise and instance-dependent label noise. Then, we conduct further experiments to verify the algorithm efficiency in Section 5.2. Finally, we compare our method with other methods in the class imbalance setting in Section 5.3.

Table 2: Results on CIFAR10 with asymmetric noise.

Method	M	$\tau = 0.2$	$\tau = 0.4$
Forward (Patrini et al., 2017)	×	90.10 ± 0.00	-
GLC (Hendrycks et al., 2018)	1k	92.46 ± 0.00	91.74 ± 0.00
GCE (Zhang & Sabuncu, 2018)	×	89.50 ± 0.30	82.30 ± 0.70
PENCIL (Yi & Wu, 2019)	×	92.43 ± 0.00	91.01 ± 0.00
JoCoR (Wei et al., 2020)	×	-	76.36 ± 0.49
DivideMix (Li et al., 2020)	×	93.40 ± 0.00	-
L2RW (Ren et al., 2018)	1k	-	86.73 ± 0.48
MW-Net (Shu et al., 2019)	1k	90.33 ± 0.61	87.54 ± 0.23
MLC (Wang et al., 2020)	0.5k	84.60 ± 0.00	79.85 ± 0.00
IEG (Zhang et al., 2020)	0.1k	96.50 ± 0.20	**95.90** ± **0.10**
FastRW (Zhang & Pfister, 2021)	×	95.00 ± 0.10	93.60 ± 0.30
Ours	×	**96.70** ± **0.12**	95.63 ± 0.41

5.1 Label Noise Setting

Generating corrupted labels. Following (Patrini et al., 2017; Xia et al., 2020), we manually corrupt the labels by constructing the noise transition matrix Q, where $Q_{ij} = P(\hat{y} = j|y = i)$ given that noise $\hat{y}$ is flipped from clean y. Assume that the matrix Q_{ij} has three types. (1) Symmetric noise, *i.e.*, $Q_{ij} = \tau/(c-1)$ for $i \neq j$ and $Q_{i,i} = 1 - \tau$, where τ is the constant noise level. (2) Asymmetric noise, the labels may make mistakes only within very similar classes, *i.e.*, $Q_{ij} = \tau$ and $Q_{ii} = 1 - \tau$. (3)

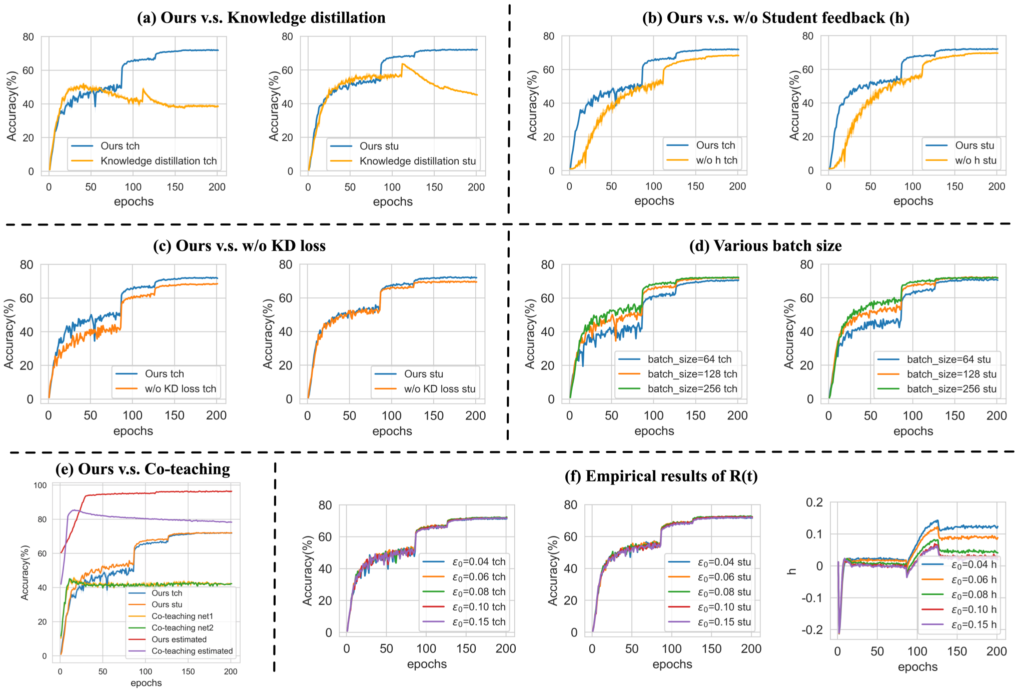

Figure 3: Ablation studies of our proposed Distilled Meta-Learning approach on CIFAR100 with 40% symmetric noise. 'tch' is short for teacher network while 'stu' is short for student network.

Instance-dependent label noise is related to the sample, *i.e.*, $Q_{ij}(X) = P(\hat{y} = j|y = i, X)$. We test the proposed method on standard benchmark datasets, CIFAR10 and CIFAR100 (Krizhevsky, 2012). For CIFAR10/100, we use three types of label noise: symmetric (Table 1), asymmetric (Table 2 & Table 3) and instance-dependent label noise (Table 4).

Table 3: Results on CIFAR100 with asymmetric noise.

Method	M	$\tau = 0.2$	$\tau = 0.4$
Forward (Patrini et al., 2017)	×	64.20 ± 0.00	-
GLC (Hendrycks et al., 2018)	1k	71.40 ± 0.00	67.73 ± 0.00
PENCIL (Yi & Wu, 2019)	×	74.70 ± 0.56	63.61 ± 0.23
JoCoR (Wei et al., 2020)	×	-	32.70 ± 0.35
L2RW (Ren et al., 2018)	1k	-	59.30 ± 0.60
MW-Net (Shu et al., 2019)	1k	64.22 ± 0.28	58.64 ± 0.47
MLC (Wang et al., 2020)	0.5k	57.22 ± 0.00	53.33 ± 0.00
Ours	×	**80.24 ± 0.66**	**69.52 ± 1.22**

Baselines. We compare the proposed method with the following representative methods: **1)** Several works apply loss correction on an estimated noise transition matrix, *e.g.*, Forward (Patrini et al., 2017), GLC (Hendrycks et al., 2018). **2)** One typical idea is to reduce the influence of noisy samples with robust loss functions, *e.g.*, GCE (Zhang & Sabuncu, 2018), Peer Loss (Liu & Guo, 2020). **3)** Another direction seeks to correct noisy labels based on model predictions, *e.g.*, PENCIL (Yi & Wu, 2019). **4)** Some methods based on sample selection to select trusted samples from training data, *e.g.*, Co-Teaching (Han et al., 2018), JoCoR (Wei et al., 2020), CORES (Cheng et al., 2021). **5)** Some approaches leverage semi-supervised techniques to learn from noisy labels, *e.g.*, DivideMix (Li et al., 2020). **6)** Other methods try to reduce the weights assigned to noisy samples, *e.g.*, MentorNet (Jiang et al., 2018), MentorMix (Jiang et al., 2020). **7)** Meta-learning-based approaches, *e.g.*, L2RW (Ren et al., 2018), MW-Net (Shu et al., 2019), MLC (Wang et al., 2020), IEG (Zhang et al., 2020), FaMUS (Xu et al., 2021), FastRW (Zhang & Pfister, 2021), Purify (Wu et al., 2021a), CAL (Zhu et al., 2021).

Implementations. For label noise experiments on CIFAR10/100, we use the WRN-28-10 (Zagoruyko & Komodakis, 2016) as it is commonly used in (Ren et al., 2018; Shu et al., 2019). We use SGD with a momentum of 0.9 with a weight decay of 5×10^{-4}, and an initial learning rate 5×10^{-2} for both teacher and student network. And the batch size is set to 128. We train the network for 300 epochs and reduce the learning rate by a factor of 10 after 100, 150 and 200 epochs. Please for details, refer to Appendix H.

Results of synthetic datasets. Table 1 shows the results for symmetric label noise on CIFAR10/100 with noise rates of $\{0.2, 0.4, 0.6\}$, respectively. We report the mean and standard deviation over three training trials using different random seeds. Our method achieves competitive results to IEG (Zhang et al., 2020) and FaMUS (Xu et al., 2021), although it marginally underperforms IEG and FAMUS, which requires additional clean data. Specifically, even under relatively high noise ratios (*e.g.*, $\tau = 0.6$ on CIFAR-100 with symmetric noise), our method has competitive classification accuracy (74.20%) without using any clean data. Table 2 and Table 3 show the results for asymmetric label noise on CIFAR10/100 with noise rates of $\{0.2, 0.4\}$, respectively. These results illustrate the effectiveness of our method on synthetic noise.

Real-world corrupted dataset. To verify the efficacy of our methods in the real-world scenario, we conduct experiments on the noisy dataset Clothing1M (Xiao et al., 2015). Specifically, for experiments on Clothing1M, we use the 1M images with noisy labels for training and 10k clean data for testing respectively. Following the previous work (Patrini et al., 2017; Tanaka et al., 2018), we used ResNet-50 (He et al., 2016) pre-trained on ImageNet. For preprocessing, we resize the image to 256×256, crop the middle 224×224 as input, and perform normalization. We used SGD with a momentum 0.9, a weight decay 10^{-4}, and an initial learning rate 10^{-3} for both teacher and student network, and batch size 64. The learning rate is divided by 10 after 5 epochs (for a total 30 epochs). The results are summarized in Table 5 which shows that the proposed method achieves the best performance.

Table 4: Results on CIFAR with instance-dependent label noise.

Method	CIFAR10		CIFAR100	
	$\tau = 0.2$	$\tau = 0.4$	$\tau = 0.2$	$\tau = 0.4$
Forward	88.08	82.67	58.95	41.68
Co-teaching	88.66	69.50	43.03	23.13
Co-teaching+	89.04	69.15	41.84	24.40
JoCoR	88.71	68.97	44.28	22.77
MentorNet	70.56	46.22	-	-
Peer Loss	89.33	81.09	59.92	45.76
CORES	89.50	82.84	61.25	41.81
CAL	92.01	84.96	69.11	63.17
Ours	**96.00**	**94.37**	**81.36**	**76.09**

Ablation study. Figure 3 shows the ablation studies of our proposed distilled meta-learning method. Notice that we use the WRN-28-2 (Zagoruyko & Komodakis, 2016) network for all experiments. From Figure 3(a), it can be seen that in the standard knowledge distillation schema, even if students learn from the soft-targets provided by teachers, *memorization effects* (Arpit et al., 2017) still occur, indicating that teachers provide error guidance for students. From the results, the accuracy was maintained at about 50%, indicating that knowledge distillation alone could not avoid the network falling into over-fitting. Figure 3(b) indicates that lacking the student's performance as feedback, also known as h, to optimize the teacher network will also affect the scale of sample selection and the final accuracy. As discussed in Section 3, by varying h, we can get the feedback of student and derive the percentage of relabeled training data against the complete training set. Figure 3(b) provides the impact of the absence of h. In Equation (2), we innovatively introduced distillation loss. In order to verify the effectiveness of this change, we eliminated the influence of distillation loss by setting its coefficient λ to 0, and the result is shown in Figure 3(c). Since Theorem 4.1 shows that the proposed method converges linearly with the number T of outer-loop meta iterations, and the convergence error decays linearly with the number B of sampled mini-batch. As shown in Figure 3(d), we studied the empirical results of the proposed meta-learning method at different batch sizes. Experimental results show that the convergence rate of the algorithm accelerates with the increase of batch size. Compared with Co-teaching based methods (Han et al., 2018; Yu et al., 2019; Wei et al., 2020) using fixed criteria to select samples, we made the selection criteria data-driven by self-learning penalty terms, and experiment Figure 3(e) proved its effectiveness. Meanwhile, the proposed selection criteria can effectively improve the accuracy of sample estimation. As it can be seen from Figure 3(e), the estimation accuracy, *i.e.*, *estimated accuracy = (# of clean labels) / (# of all selected labels)*, of our method increases steadily. To show the impact of ϵ_0 on $R(t)$ in Algorithm 1, we vary $\epsilon_0 = \{0.04, 0.06, 0.08, 0.10, 0.15\}$ in Figure 3(f). Note that, ϵ_0 cannot be zero. In this case, since no samples are selected, there is no gradient back-propagation, and the optimization will stop.

Table 5: Comparison with state-of-the-art methods in test accuracy (%) on Clothing1M.

Method	Test Accuracy
PENCIL (Yi & Wu, 2019)	73.49
MLNT (Li et al., 2019)	73.47
MW-Net (Shu et al., 2019)	73.72
DivideMix (Li et al., 2020)	74.76
JoCoR (Wei et al., 2020)	70.30
MLC (Wang et al., 2020)	71.06
FaMUS (Xu et al., 2021)	74.43
Ours	**75.23**

Hyper-parameter analysis. We evaluate the impact of four hyper-parameters on the CIFAR100 dataset with 40% symmetric noise. Due to the limited space, the results are shown in Appendix H. As shown in Figure 6, λ controls the relative importance of $\mathcal{L}_{direct}$ and $\mathcal{L}_{KD}$. α is used in the extended Algorithm 2 as an unsupervised loss coefficient to control the contribution of unselected samples. The third figure in Figure 6 reports the effect of distillation temperature ξ. The fourth figure in Figure 6 reports the effect of the teacher's label smoothing parameter. We use the controlled variable method to observe the effect by varying the selected hyper-parameters and freezing the remaining hyper-parameters. The biggest change by sweeping these hyper-parameters is 2%, indicting insensitivity to hyper-parameters.

5.2 Algorithm efficiency

Table 6 shows the results on the CIFAR10 dataset with 40% symmetric noise, where the "Time" column lists the average running time (in millisecond) per training iteration on a single NVIDIA 3090 GPU. For fair and consistent comparisons, we use the WRN-28-2 network architecture for all methods, and train with a mini-batch size of 100. The teacher network obtains 0.25 GFLOPs as well as the student network. Experimental results show that our method can significantly reduce the training time compared with other meta-learning algorithms (Ren et al., 2018; Shu et al., 2019; Zhang et al., 2020; Xu et al., 2021), while still maintaining comparable generalization performance or achieving better generalization performance.

Table 6: Training complexity by time per second.

Method	Time(ms)	Acc.($\%$)
L2RW	903	86.92
MW-Net	954	89.59
IEG	1416	92.80
L2RW + FaMUS	266	87.60
MW-Net + FaMUS	304	90.50
Ours (Second-order)	321	93.01
Ours (First-order)	160	92.72

5.3 Long-tailed Imbalance Setting

Table 7: Test accuracy (%) of ResNet32 (He et al., 2016) on long-tailed CIFAR10 and CIFAR100. M represents the number of additional balanced data required for the method.

Dataset	CIFAR10				CIFAR100			
Imb. ratio	M	200	50	10	M	200	50	10
SoftMax	×	65.68	74.81	86.39	×	34.84	43.85	55.71
Focal Loss (Lin et al., 2017)	×	65.29	76.71	86.66	×	35.62	44.32	55.78
Class-Balanced	×	68.89	79.27	87.49	×	36.23	45.32	57.99
L2RW (Ren et al., 2018)	1k	66.51	78.93	85.19	1k	33.38	44.44	53.73
MW-Net (Shu et al., 2019)	1k	68.91	80.06	87.84	1k	37.91	46.74	58.46
FaMUS (Xu et al., 2021)	×	67.76	79.17	87.40	×	35.44	42.57	55.45
Ours	×	**70.02**	**81.43**	**88.67**	×	**36.78**	**46.75**	**60.21**

Setup and comparison methods. We test our method on long-tailed CIFAR (Krizhevsky et al., 2009) benchmarks, with different imbalance ratios as defined by (Cui et al., 2019). Long-Tailed CIFAR are created by reducing the number of training instances per class according to an exponential function $n = n_i \mu^i$, where i is the class index, n_i the original number of training images and $\mu \in (0, 1)$. We modify some training configurations as we use for label noise experiments. Following recent methods (Cao et al., 2019; Kang et al., 2020) commonly apply deferred balancing, which trains model in the supervised manner to learn representations and then applies re-balancing methods to fine-tune the model. Overall, our proposed distilled meta-learning approach achieves competitive results to other methods, even though MW-Net requires additional balanced data.

6 Conclusion

In this paper, we present an efficiently distilled meta-learning approach to overcome the two aforementioned problems: 1) removing the dependence of prior unbiased set and 2) improving training efficiency. The proposed method dynamically estimates unbiased samples in training data and leverages meta-learning to refine the deep networks. We conduct extensive experiments to demonstrate its effectiveness on both noisy labels and long-tailed recognition benchmarks.

REFERENCES

Devansh Arpit, Stanislaw Jastrzkebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S. Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, and Simon Lacoste-Julien. A closer look at memorization in deep networks. In *International Conference on Machine Learning (ICML)*, pp. 233–242, 2017.

Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in neural information processing systems (NIPS)*, volume 32, 2019.

Mathilde Caron, Hugo Touvron, Ishan Misra, Herve Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

Can Chen, Shuhao Zheng, Xi Chen, Erqun Dong, Xue (Steve) Liu, Hao Liu, and Dejing Dou. Generalized dataweighting via class-level gradient manipulation. In *Advances in Neural Information Processing Systems (NIPS)*, volume 34, pp. 14097–14109, 2021.

Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. In *International Conference on Learning Representations (ICLR)*, 2021.

Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9268–9277, 2019.

Pedro Domingos. A unified bias-variance decomposition. In *International Conference on Machine Learning (ICML)*, pp. 231–238, 2000.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pp. 1082–1092, 2020.

Tongtong Fang, Nan Lu, Gang Niu, and Masashi Sugiyama. Rethinking importance weighting for deep learning under distribution shift. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11996–12007. Curran Associates, Inc., 2020.

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

Chelsea Finn, Kelvin Xu, and Sergey Levine. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International Conference on Machine Learning (ICML)*, volume 97, pp. 1920–1930, 2019.

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 (AAAI)*, 2017.

J. Goldberger and E. Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *International Conference on Learning Representations (ICLR)*, 2017.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems (NeurIPS)*, pp. 8527–8537, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Dan Hendrycks, Mantas Mazeika, Duncan Wilson, and Kevin Gimpel. Using trusted data to train deep networks on labels corrupted by severe noise. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee. In *International Conference on Learning Representations (ICLR)*, 2020.

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

Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. Beyond synthetic noise: Deep learning on controlled noisy labels. In *International Conference on Machine Learning (ICML)*, pp. 4804–4815, 2020.

Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1gRTCVFvB.

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

Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 2012.

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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 (CVPR)*, 2019.

Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2020.

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

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 20331–20342, 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 (ICML)*, pp. 6226–6236, 2020.

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 (ICML)*, pp. 6543–6553, 2020.

Eran Malach and Shai Shalev-Shwartz. Decoupling when to update from how to update. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 960–970, 2017.

Aditya Menon, Brendan Van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from corrupted binary labels via class-probability estimation. In *International Conference on Machine Learning (ICML)*, pp. 125–134, 2015.

Aditya Krishna Menon, Ankit Singh Rawat, Sashank J. Reddi, and Sanjiv Kumar. Can gradient clipping mitigate label noise? In *International Conference on Learning Representations (ICLR)*, 2020.

Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

Kento Nishi, Yi Ding, Alex Rich, and Tobias Hollerer. Augmentation strategies for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8022–8031, 2021.

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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1944–1952, 2017.

Aniruddh Raghu, Maithra Raghu, Samy Bengio, and Oriol Vinyals. Rapid learning or feature reuse? towards understanding the effectiveness of maml. In *International Conference on Learning Representations (ICLR)*, 2020.

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

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

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

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Machine Learning (ICLR)*, 2019.

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 (ICCV)*, pp. 322–330, 2019.

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 (CVPR)*, 2020.

Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. Learning to purify noisy labels via meta soft label corrector. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pp. 10388–10396, 2021a.

Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. Learning to purify noisy labels via meta soft label corrector. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021b.

Xiaobo Xia, T. Liu, N. Wang, B. Han, C. Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6838–6849, 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. In *Advances in neural information processing systems (NeurIPS)*, 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2691–2699, 2015.

Qizhe Xie, Zihang Dai, Eduard H. Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

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

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

Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor W Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning (ICML)*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Proceedings of the British Machine Vision Conference (BMVC)*, pp. 87.1–87.12, 2016.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.

HaiYang Zhang, XiMing Xing, and Liang Liu. Dualgraph: A graph-based method for reasoning about label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9654–9663, 2021.

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

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

Zizhao Zhang and Tomas Pfister. Learning fast sample re-weighting without reward data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 725–734, 2021.

Zizhao Zhang, Han Zhang, Sercan O. Arik, Honglak Lee, and Tomas Pfister. Distilling effective supervision from severe label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Evgenii Zheltonozhskii, Chaim Baskin, Avi Mendelson, Alex M Bronstein, and Or Litany. Contrast to divide: Self-supervised pre-training for learning with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1657–1667, 2022.

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

Zhaowei Zhu, Tongliang Liu, and Yang Liu. A second-order approach to learning with instance-dependent label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10113–10123, 2021.

Luisa M Zintgraf, Kyriacos Shiarlis, Vitaly Kurin, Katja Hofmann, and Shimon Whiteson. CAML: Fast context adaptation via meta-learning. In *International Conference on Learning Representations (ICLR)*, 2019.

APPENDIX

A DERIVATION OF THE META UPDATE RULE.

Understanding that these notations and conventions might cause confusions, in the derivation below. In expectation, the new parameter of student is:

$$\bar{\theta}'_S = \mathbb{E}_{\bar{y}\sim P(\cdot|x;\theta_T)}\left[\theta_S - \gamma \nabla_{\theta_S}\mathcal{L}\left(\bar{y}, \mathcal{F}(x \mid \theta_S)\right)\right] \tag{11}$$

where $\theta_S \in \mathbb{R}^{|S|\times 1}$. We will update the θ_T to minimize the knowledge distillation loss on student:

$$\frac{\partial \mathcal{R}}{\partial \theta_T} = \frac{\partial}{\partial \theta_T}\mathcal{L}\left(\tilde{y}, \mathcal{F}\left(x \mid \bar{\theta}'_S\right)\right) \tag{12}$$

where $\theta_T \in \mathbb{R}^{|T|\times 1}$.

We now present the derivation.

$$\frac{\partial \mathcal{R}}{\partial \theta_T} = \frac{\partial}{\partial \theta_T}\mathcal{L}\left(\tilde{y}, \mathcal{F}(\tilde{x} \mid \bar{\theta}'_S)\right) = \left.\frac{\partial \mathcal{L}\left(\tilde{y}, \mathcal{F}(\tilde{x} \mid \bar{\theta}'_S)\right)}{\partial \theta_S}\right|_{\theta_S=\bar{\theta}'_S} \cdot \frac{\partial \bar{\theta}'_S}{\partial \theta_T} \tag{13}$$

where first factor has dimension $1 \times |S|$, and second factor has dimension $|S| \times |T|$. The first factor in Equation (13) can be simply computed via back-propagation. We now focus on the second term. We have:

$$\frac{\partial \bar{\theta}'_S}{\partial \theta_T} = \frac{\partial}{\partial \theta_T}\mathbb{E}_{\bar{y}\sim P(\cdot|x;\theta_T)}\left[\theta_S - \gamma \cdot \nabla_{\theta_S}\mathcal{L}\left(\bar{y}, \mathcal{F}(\cdot \mid \theta_S)\right)\right] \tag{14}$$

where the Jacobian of $\mathcal{L}\left(\bar{y}, \mathcal{F}\left(x \mid \theta_S\right)\right)$ needs to be transposed to $|S| \times 1$ to match the dimensions of θ_S. Since θ_S in Equation (14) does not depend on θ_T, we can leave it out of subsequent derivations. Also, to simplify notations, let us define the gradient:

$$g_{\mathcal{F}(\bar{y}|\theta_S)} = \left(\frac{\partial \mathcal{L}\left(\bar{y}, \mathcal{F}(\cdot \mid \theta_S)\right)}{\partial \theta_S}\right)^{\top} \tag{15}$$

where $g_{\mathcal{F}(\bar{y}|\theta_S)} \in \mathbb{R}^{|S|\times 1}$. The the Equation (14) becomes:

$$\frac{\partial \bar{\theta}'_S}{\partial \theta_T} = -\gamma \cdot \frac{\partial}{\partial \theta_T}\mathbb{E}_{\bar{y}\sim P(\cdot|x;\theta_T)}\left[g_{\mathcal{F}(\bar{y}|\theta_S)}\right] \tag{16}$$

Since $g_{\mathcal{F}(\bar{y}|\theta_S)}$ has no dependency on θ_T,

$$\begin{aligned}\frac{\partial \bar{\theta}_S^{(t+1)}}{\partial \theta_T} &= -\gamma \cdot \frac{\partial}{\partial \theta_T}\mathbb{E}_{\bar{y}\sim P(\cdot|x;\theta_T)}\left[g_{\mathcal{F}(\bar{y}|\theta_S)}\right] \\ &= -\gamma \cdot \mathbb{E}_{\bar{y}\sim P(\cdot|x;\theta_T)}\left[g_{\mathcal{F}(\bar{y}|\theta_S)} \cdot \frac{\partial \log P\left(\bar{y} \mid x;\theta_T\right)}{\partial \theta_T}\right] \\ &= \gamma \cdot \mathbb{E}_{\bar{y}\sim P(\cdot|x;\theta_T)}\left[g_{\mathcal{F}(\bar{y}|\theta_S)} \cdot \frac{\partial \mathcal{L}(\bar{y}, \mathcal{F}(\cdot \mid \theta_T))}{\partial \theta_T}\right]\end{aligned} \tag{17}$$

where $\frac{\partial \mathcal{L}(\tilde{y}, \mathcal{F}(\cdot|\theta_T))}{\partial \theta_T} \in \mathbb{R}^{1\times|T|}$, Now, we can substitute Equation (17) into Equation (13) to obtain:

$$\begin{aligned} \frac{\partial \mathcal{R}}{\partial \theta_T} &= \left.\frac{\partial \mathcal{L}\left(\tilde{y}, \mathcal{F}(\tilde{x} \mid \bar{\theta}'_S)\right)}{\partial \theta_S}\right|_{\theta_S=\bar{\theta}'_S} \cdot \frac{\partial \bar{\theta}'_S}{\partial \theta_T} \\ &= \gamma \cdot \nabla_{\bar{\theta}'_S} \mathcal{L}_u\left(\theta_S\right) \cdot \mathbb{E}_{\bar{y}\sim P(\cdot|x;\theta_T)}\left[g_{\mathcal{F}(\bar{y}|\theta_S)} \cdot \frac{\partial \mathcal{L}(\tilde{y}, \mathcal{F}(x \mid \theta_T))}{\partial \theta_T}\right] \end{aligned} \tag{18}$$

where $\left.\frac{\partial \mathcal{L}\left(\tilde{y}, \mathcal{F}(\tilde{x}|\bar{\theta}'_S)\right)}{\partial \theta_S}\right|_{\theta_S=\bar{\theta}'_S} := \nabla_{\bar{\theta}'_S} \mathcal{L}_u\left(\theta_S\right)$.

Finally, we use Monte Carlo approximation for every term in Equation (18) using the sampled $\bar{y}$. In particular, we approximate $\bar{\theta}'_S$ with the parameter obtained from θ_S via updating the student parameter on $\bar{y} \sim P(\cdot \mid x; \theta_T)$, i.e., $\theta'_S = \theta_S - \gamma \cdot \nabla_{\theta_S} \ell\left(\bar{y}, \mathcal{F}(\cdot \mid \theta_S)\right)$, and approximate the expected value in the second term with $\bar{y}$. With these approximation, we obtain the gradient:

$$\begin{aligned} \nabla_{\theta_T} \mathcal{L}(\theta_T, \theta_S) &= \gamma \cdot \frac{\partial \mathcal{L}\left(\tilde{y}, \mathcal{F}(\tilde{x} \mid \theta'_S)\right)}{\partial \theta_S} \cdot \left(\frac{\partial \mathcal{L}\left(\bar{y}, \mathcal{F}(x \mid \theta'_S)\right)}{\partial \theta_S}\right)^{\top} \cdot \frac{\partial \mathcal{L}\left(\bar{y}, \mathcal{F}\left(x; \theta_T\right)\right)}{\partial \theta_T} \\ &= \gamma \cdot h \cdot \nabla_{\theta_T} \mathcal{L}(\bar{y}, \mathcal{F}(x; \theta_T)) \end{aligned} \tag{19}$$

where $\frac{\partial \mathcal{L}\left(\tilde{y}, \mathcal{F}(\tilde{x}|\theta'_S)\right)}{\partial \theta_S} \in \mathbb{R}^{1\times|S|}$, and $\left(\frac{\partial \mathcal{L}\left(\bar{y}, \mathcal{F}(x|\theta'_S)\right)}{\partial \theta_S}\right)^{\top} \in \mathbb{R}^{|S|\times 1}$. The product of these two terms is a scalar, which is recorded as h. We rewrite the computation of this term: $\tilde{h} = \left[\nabla_{\theta'_S} \mathcal{L}\left(\tilde{y}, \mathcal{F}\left(\tilde{x}; \theta'_S\right)\right)\right]^{\top} \cdot \nabla_{\theta_S} \mathcal{L}\left(\bar{y}, \mathcal{F}\left(x; \theta_S\right)\right)$.

B Pseudo Code

In this section, we present pseudo code for a complete version of our proposed Distilled Meta-Learning method. A limitation of the sample-selection-based method is that the examples that are selected tend to be easier, and the subset of selected examples may not be rich enough to generalize effectively to held-out data (Liu et al., 2020). This suggests that in addition to estimate unbiased samples from the noisy data, it is also important to estimate the correct labels via re-labeling process. We informally call this process as estimation of "Data cleaning", which are two major information for constructing supervised training. This extension is very simple. We follow (Xie et al., 2020) and give a new label y_{new} to the unselected samples via the teacher's prediction. In particular, we extend the proposed approach to include re-labeling loss in the optimization objectives of the teacher network. In order to facilitate the study of its influence, we set the coefficient of re-labeling loss as α, and the experimental details can be found in Figure 7. We emphasize that the re-labeling objective is applied on the teacher, while the student still only learns from the teacher. The pseudo code can be found in Algorithm 2.

C Risk Analysis

In this section, we analyze the risk of student network learning by knowledge distillation in Distilled Meta-Learning framework. Similar to (Li et al., 2017), we define the expected risk $\mathcal{R}_y$ associated with the unreliable label sampled from $\mathcal{D}$: $\mathcal{R}_{\hat{y}} = \mathbb{E}_{\mathcal{D}}\left[\ell(\hat{y}, y^*)\right]$. where y^* is the unknown ground-truth label, and $\mathbb{E}$ denotes the expectation over $\mathcal{D}$. The random variable $\hat{y}$ denotes the unreliable label corrupted from the y^*. Although $\mathcal{R}_{\hat{y}}$ does not relate directly with the final accuracy of the classifier, it is an indicator of the level of noise seen by the training process, which implicitly affects the final performance. Then, we show that the risk of using Equation (1) can be smaller than using either the full noisy labels or only the partial clean labels.

Proof. First, we rewrite a risk $\mathcal{R}_{\hat{y}}$ associated with the unreliable label $\hat{y}$ for the tractability of analysis:

$$\mathcal{R}_y = \mathbb{E}_{\mathcal{D}}\left[\|\hat{y} - y^*\|^2\right] \tag{20}$$

Algorithm 2 Pseudo Code for Our Methods

```
Input: Train set $\mathcal{D} = \{(x_i, y_i)|_{i=1}^{N}\}$, the teacher net $\mathcal{F}(\cdot \mid \theta_T)$, the student net $\mathcal{F}(\cdot \mid \theta_S)$, teacher and student net learning rate $\gamma$, $\eta$, epoch $T$, fixed $\epsilon_0$.
Initialize $\theta_T^{(0)}$ and $\theta_S^{(0)}$.
Shuffle training set $\mathcal{D}$.
for $t = 0$ to $T$ do
    Fetch mini-batch $\mathcal{D}_n$ from $\mathcal{D}$.
    Obtain $\widetilde{\mathcal{D}} = \arg\min_{D':|D'_n| \geq R(t)|\mathcal{D}_n|} [\mathcal{L}(\mathcal{D}; \theta_T; \theta_S)]$
    where $\widetilde{\mathcal{D}} = \{(\tilde{x}_i, \tilde{y}_i)|_{i=1}^{R(t)|\mathcal{D}_n|}\}$
    Sample a pseudo label $\bar{y} \sim P(\cdot \mid x_i; \theta_T)$.
    Compute one step gradient of knowledge distillation:
    $\hat{\theta}_S^{t+1} = \theta_S^{(t)} - \gamma \nabla_{\theta_S} \ell_{ce}(\bar{y}, \mathcal{F}(x; \theta_S)) |_{\theta_S = \theta_S^{(t)}}$
    Compute the student's feedback:
    $h = \eta \cdot \left[ \left( \nabla_{\theta'_S} \ell_{ce} \left( \tilde{y}, \mathcal{F}\left( \tilde{x}; \theta_S^{(t+1)} \right) \right) \right)^{\top} \cdot \nabla_{\theta_S} \ell_{ce} \left( \bar{y}, \mathcal{F}\left( x; \hat{\theta}_S^{t+1} \right) \right) \right]$
    Compute $g_T^{(T)} = h \cdot \nabla_{\theta'_T} \ell_{ce}(\bar{y}, f_t(x; \theta_T)) |_{\theta_T = \theta_T^{(t)}}$ { // the teacher's gradient on student's feedback}
    Compute $g_{T,\widetilde{\mathcal{D}}}^{(t)} = \nabla_{\theta'_T} \ell_{ce}(\tilde{y}, \mathcal{F}(\tilde{x}; \theta_T)) |_{\theta_T = \theta_T^{(t)}}$ { // the teacher's gradient on unbiased data}
    Compute $g_{T,\mathcal{D}}^{(t)} = \nabla_{\theta'_T} \ell_{ce}(y_{new}, \mathcal{F}(\tilde{x}; \theta_T)) |_{\theta_T = \theta_T^{(t)}}$ { // the teacher's gradient on refined labels}
    update teacher:
    $\theta_T^{t+1} = \theta_T^{t} - \eta \cdot \left( g_T^{(t)} + g_{T,\widetilde{\mathcal{D}}}^{(t)} + g_{T,\mathcal{D}}^{(t)} \right)$
    update $R_t(c)$
end for
return $\theta_S^{(T)}$
```

Ideally, we would like to define the risk according to the training loss, but ℓ_2 distance is used for the tractability of analysis. The risk of using labels corrupted by noise as $\hat{y} \sim P_{\mathcal{D}}(\hat{y} \mid (x, y^*))$ is quantified by the following residual term. Consider our teacher model trained from an unbiased dataset $\tilde{\mathcal{D}}$, the expected prediction error can be decomposed into the variance term and the bias term (Domingos, 2000):

$$\mathbb{E}_{\tilde{\mathcal{D}}}[\ell(\tilde{y}, y^*)] = \ell(\bar{s}, y^*) + \mathbb{E}_{\tilde{\mathcal{D}}}[\ell(\tilde{y}, \bar{s})] \tag{21}$$

where $\ell(\cdot, \cdot)$ is the loss function, and $\bar{s}(x)$ is called the "main" prediction, defined according to the loss function. For the squared loss $\bar{s}(x) = \text{average}_{\tilde{\mathcal{D}}}(\mathcal{F}(x))$. For simplicity of proving Equation (10), we use the squared loss. Since we are training a over parameterized CNN, we can make a reasonable assumption that the bias term $\ell(\bar{s}, y^*)$ is close to zero. Therefore, we have $\ell(\bar{s}, y^*) \approx 0 \Rightarrow \bar{s} \approx y^*$.

$$\mathbb{E}_{\tilde{\mathcal{D}}}\left(\|\tilde{y} - y^*\|^2\right) \approx \mathbb{E}_{\tilde{\mathcal{D}}}\left(\|\tilde{y} - \bar{s}\|^2\right) \triangleq \mathcal{R}_s \tag{22}$$

The label corruption process is unknown, but we can assume that it is independent of the model variance. This leads to:

$$\begin{aligned}
\mathbb{E}_{\tilde{\mathcal{D}}}\left[(\hat{y} - y^*)^{\top}(\tilde{y} - y^*)\right] &= \left(\mathbb{E}_{\tilde{\mathcal{D}}}[\hat{y} - y^*]\right)^{\top} \mathbb{E}_{\tilde{\mathcal{D}}}[\tilde{y} - y^*] \\
&= \left(\mathbb{E}_{\tilde{\mathcal{D}}}[\hat{y} - y^*]\right)^{\top} \mathbb{E}_{\tilde{\mathcal{D}}}[\tilde{y} - \bar{s}] \\
&= \left(\mathbb{E}_{\tilde{\mathcal{D}}}[\hat{y} - y^*]\right)^{\top} \mathbf{0} = 0
\end{aligned} \tag{23}$$

where **0** denotes a zero vector. Now, we are ready to show Equation (10):

$$\begin{aligned}
\mathcal{R}_{\tilde{y}^{\lambda}} &= \mathbb{E}_{\tilde{\mathcal{D}}}\left[\|\tilde{y}^{\lambda} - y^*\|^2\right] \\
&= \mathbb{E}_{\tilde{\mathcal{D}}}\left[\|(1-\lambda)\tilde{y} + \lambda\tilde{y} - y^*\|^2\right] \\
&= \mathbb{E}_{\tilde{\mathcal{D}}}\left[\|(1-\lambda)(y - y^*) + \lambda(\bar{y} - y^*)\|^2\right] \\
&= (1-\lambda)^2 \mathcal{R}_{\hat{y}} + \lambda^2 \mathcal{R}_{\tilde{y}}
\end{aligned} \tag{24}$$

When $\lambda = \frac{\mathcal{R}_{\hat{y}}}{\mathcal{R}_{\hat{y}}+\mathcal{R}_{\tilde{y}}}$, $\mathcal{R}_{\bar{y}^\lambda}$ reaches its minimum,

$$\min_{\lambda} \mathcal{R}_{\bar{y}^\lambda} = \frac{\mathcal{R}_{\hat{y}}\mathcal{R}_{\tilde{y}}}{\mathcal{R}_{\hat{y}}+\mathcal{R}_{\tilde{y}}} \tag{25}$$

The proof is completed. □

D Convergence Proof of Our Method

This section provides the proofs for Theorems Theorem D.4 and Theorem 4.2 in the paper.

D.1 Basic Assumptions

We state several standard assumptions for the analysis in the proposed distilled meta-learning approach.

Assumption D.1. *For each mini-batch, the loss function $\ell_S(\cdot)$ and $\ell_{T_i}(\cdot)$ in Equation (1) satisfy*

1. $\ell_S(\cdot)$, $\ell_{T_i}(\cdot)$ *are bounded below, i.e.,* $\inf_{\theta_S \in \mathbb{R}^{|S|}} \ell_S(\theta_S) > -\infty$ *and* $\inf_{\theta_T \in \mathbb{R}^{|T|}} \ell_{T_i}(\theta_T) > -\infty$.
2. *Gradients $\nabla \ell_S(\cdot)$ and $\nabla \ell_{T_i}(\cdot)$ are L-Lipschitz continuous, i.e., for any $w = \theta_S, u = \theta_T \in \mathbb{R}^d$*
$$\|\nabla \ell_S(w) - \nabla \ell_S(u)\| \leq L\|w-u\| and \|\nabla \ell_{T_i}(w) - \nabla \ell_{T_i}(u)\| \leq L\|w-u\|.$$
3. *Hessians $\nabla^2 \ell_S(\cdot)$ and $\nabla^2 \ell_{T_i}(\cdot)$ are ρ-Lipschitz continuous, i.e., for any $w = \theta_S, u = \theta_T \in \mathbb{R}^d$*
$$\|\nabla^2 \ell_S(w) - \nabla^2 \ell_S(u)\| \leq \rho\|w-u\| and \|\nabla^2 \ell_{T_i}(w) - \nabla^2 \ell_{T_i}(u)\| \leq \rho\|w-u\|.$$

The following assumption provides two conditions $\nabla \ell_S(\cdot)$ and $\nabla \ell_{T_i}(\cdot)$.

Assumption D.2. *For all $w \in \mathbb{R}^d$, gradients $\nabla \ell_S(w)$ and $\nabla \ell_{T_i}(w)$ satisfy*

1. $\nabla \ell_{T_i}(\cdot)$ *has a bounded variance, i.e., there exists a constant $\sigma > 0$ such that*
$$\mathbb{E}_i \|\nabla \ell_{T_i}(w) - \nabla \ell_T(w)\|^2 \leq \sigma^2,$$
where $\nabla \ell_T(\cdot) = \mathbb{E}_i[\nabla \ell_T(\cdot)]$.
2. *For each $i \in \mathcal{I}$, there exists a constant $b_i > 0$ such that $\|\nabla \ell_S(w) - \nabla \ell_{T_i}(w)\| \leq b_i$.*

Instead of imposing a bounded variance condition on the stochastic gradient $\nabla \ell_S(\cdot)$, we alternatively assume the difference $\|\nabla \ell_S(w) - \nabla \ell_{T_i}(w)\|$ to be upper-bounded by a constant, which is more reasonable because sample sets S_i and T_i are often sampled from the same distribution and share certain statistical similarity. We note that the second condition also implies $\|\nabla \ell_S(w)\| \leq \|\nabla \ell_{T_i}(w)\| + b_i$ which is weaker than the bounded gradient assumption made in papers such as MAML (Finn et al., 2017). It is worthwhile mentioning that the second condition can be relaxed to $\|\nabla \ell_S(w)\| \leq c_i\|\nabla \ell_{T_i}(w)\| + b_i$ for a constant $c_i > 0$. Without the loss of generality, we consider $c_i = 1$ for simplicity.

D.2 Properties of Meta Gradient

We develop several important properties of the meta gradient. The following proposition characterizes a Lipschitz property of the gradient of the objective function

$$\nabla \mathcal{L}(w) = \mathbb{E}_{i \sim \mathcal{D}} \prod_{j=0}^{N-1} (I - \gamma \nabla^2 \ell_S(\tilde{w}_j^i)) \nabla \ell_{T_i}(\tilde{w}_N^i), \tag{26}$$

where the weights $\tilde{w}_j^i \in \mathcal{I}$. In general, $j = 0, \ldots, N$ are given by the gradient descent steps in Equation (4) and here we conduct a distillation objective at inner loop optimization to simplify N to 1.

Theorem D.3. *Suppose that Assumption D.1 and Assumption D.2 hold. Then, for any $w, u \in \mathbb{R}^d$, we have*

$$\|\nabla \mathcal{L}(w) - \nabla \mathcal{L}(u)\| \leq L_w\|w-u\|, L_w = (1+\gamma L)^2 L + C_b b + C_{\mathcal{L}} \mathbb{E}\|\nabla \ell_{T_{i(w)}}\|$$

where $b = \mathbb{E}_i[b_i]$ *and* $C_b, C_{\mathcal{L}}{>}0$ *are constants given by*

$$\begin{aligned} C_b &= (\gamma\rho + \frac{\rho}{L}(1+\gamma L))(1+\gamma L)^2 = \frac{\rho}{L}(2\rho\gamma)(1+\gamma L)^2, \\ C_{\mathcal{L}} &= (\gamma\rho + \frac{\rho}{L}(1+\gamma L))(1+\gamma L)^2 = \frac{\rho}{L}(2\rho\gamma)(1+\gamma L)^2 \end{aligned} \tag{27}$$

Proof. By the definition of $\nabla_{\mathcal{L}(\cdot)}$, we have

$$\begin{aligned} \|\nabla\mathcal{L}_i(w) - \nabla\mathcal{L}_i(u)\| \leq & \|(I - \gamma\nabla^2\ell_S(\tilde{w}_0))\nabla\ell_{T_i}(\tilde{w}^i) - (I - \gamma\nabla^2\ell_S(\tilde{u}_0))\nabla\ell_{T_i}(\tilde{w}^i)\| \\ & + \|(I - \gamma\nabla^2\ell_S(\tilde{u}_0))\nabla\ell_{T_i}(\tilde{w}^i) - (I - \gamma\nabla^2\ell_S(\tilde{u}_0))\nabla\ell_{T_i}(\tilde{u}^i)\| \\ \leq & \|(I - \gamma\nabla^2\ell_S(\tilde{u})) - (I - \gamma\nabla^2\ell_S(u))\|\|\nabla\ell_{T_i}(\tilde{w}^i)\| \\ & + (1+\gamma L)\|\nabla\ell_{T_i}(\tilde{w}^i) - \nabla\ell_{T_i}(\tilde{u}^i)\| \end{aligned} \tag{28}$$

where $\|(I - \gamma\nabla^2\ell_S(\tilde{u})) - (I - \gamma\nabla^2\ell_S(u))\| := A$. And we next upper-bound A in the above inequality. Specifically, we have

$$\begin{aligned} A \leq & \|(I - \gamma\nabla^2\ell_S(\tilde{w})) - (I - \gamma\nabla^2\ell_S(\tilde{w}))(I - \gamma\nabla^2\ell_S(\tilde{u}))\| \\ & + \|(I - \gamma\nabla^2\ell_S(\tilde{w}))(I - \gamma\nabla^2\ell_S(\tilde{u})) - (I - \gamma\nabla^2\ell_S(\tilde{u}))\| \\ & \leq ((1+\gamma L)\gamma\rho + \frac{\rho}{L}(a+\gamma L)[(1+\gamma L) - 1])\|w - u\| \end{aligned} \tag{29}$$

Combining Equation (28) and Equation (29) yields

$$\begin{aligned} \|\nabla\mathcal{L}_i(w) - \nabla\mathcal{L}_i(u)\| \leq & ((a+\gamma L)\gamma\rho + \frac{\rho}{L}(a+\gamma L)((1+\gamma L) - 1))\|w-u\|\|\nabla\ell_{T_i}(w^i)\| \\ & + (1+\gamma L)L\|\tilde{w}^i - \tilde{u}^i\| \end{aligned} \tag{30}$$

To upper-bound $\|\nabla\ell_{T_i}(w^i)\|$ above, following the Assumption D.2, using the mean value theorem, we have

$$\begin{aligned} \|\nabla\ell_{T_i}(w^i)\| = \|\nabla\ell T_i(w - \gamma\nabla\ell_S(\tilde{w}))\| \leq & \|\nabla\ell_{T_i}(w)\| + \gamma L(1+\gamma L)\nabla\ell_S(\tilde{w}) \\ \leq & (1+\gamma L)\|\nabla\ell_{T_i}(w)\| + \gamma L + b_i \end{aligned} \tag{31}$$

then, we have

$$\|\tilde{w}^i - \tilde{u}^i\| \leq (1+\gamma L)\|w - u\| \tag{32}$$

Combining the above equation yields

$$\begin{aligned} \|\nabla\mathcal{L}_i(w) - \nabla\mathcal{L}_i(u)\| \leq & \left[(1+\gamma L)\gamma\rho + \frac{\rho}{L}\gamma(1+\gamma L)\right](1+\gamma L)\|\nabla\ell_{T_i}(w)\|\|w-u\| \\ & + \left[(1+\gamma L)\gamma\rho + \frac{\rho}{L}\gamma(1+\gamma L)\right](1+\gamma L)b_i\|w-u\| \\ & + (1+\gamma L)^2 L\|w-u\|, \end{aligned} \tag{33}$$

which, in conjunction with C_b and $C_{\mathcal{L}}$ given in Equation (27), yields

$$\|\nabla\mathcal{L}_i(w) - \nabla\mathcal{L}_i(u)\| \leq \left[(1+\gamma L)^2 L + C_b + C_{\mathcal{L}}\|\nabla\ell_{T_i}(w)\|\right]\|w-u\|. \tag{34}$$

Based on the above inequality and Jensen's inequality, we finish the proof.

□

Theorem D.3 shows that $\nabla\mathcal{L}(w)$ has a Lipschitz parameter $\mathcal{L}(w)$.

Similarly to analysis of MAML Fallah et al. (2020), we use the following construction

$$\hat{L}_{w_t} = (1+\gamma L)^2 L + C_b b + \frac{C_{\mathcal{L}}}{|B_t'|} \sum_{i \in B_k'} \|\nabla \ell_{T_i}(w_t)\| \tag{35}$$

at the t-th outer-stage iteration to approximate $\hat{L}_{w_t}$, where $B_t' \subset \mathcal{I}$ is chosen independently from B_t. It can be verified that the gradient given in Equation (26) is an unbiased estimate of $\hat{L}_{w_t}$. Thus, our next step is to upper-bound the second moment of Equation (26).

Theorem D.4. *Suppose that Assumption D.1 and Assumption D.2 hold, and define constants*

$$\begin{aligned} A_w &= \frac{4(1+\gamma L)^4}{2-(1+\gamma L)^4}, \\ A_b &= \frac{4(1+\gamma L)^8}{2-(1+\gamma L)^4}(\sigma+b)^2 + 2(1+\gamma)^4(\sigma^2+\tilde{b}), \end{aligned} \tag{36}$$

where $\tilde{b} = \mathbb{E}_{i\sim\mathcal{D}}[b_i^2]$. Then if $\gamma < \frac{\sqrt{2}}{L}$, then conditioning on w_t, we have

$$\mathbb{E}\|\hat{G}_i(w_t)\|^2 \le A_w \|\nabla \mathcal{L}(w_k)\|^2 + A_b. \tag{37}$$

Proof. Conditioning on w_t, we have

$$\mathbb{E}\|\hat{G}_i(w_t)\|^2 = \mathbb{E}\|(I - \gamma \nabla^2 \ell_S(w_t)) \nabla \ell_{T_i}(w_t^i)\| \le (a+\gamma L)^2 \mathbb{E}\|\nabla \ell_{T_i}(w_t^i)\|, \tag{38}$$

then, we have

$$\begin{aligned} \mathbb{E}\|\hat{G}_i(w_t)\|^2 &\le 2(1+\gamma L)^4 \mathbb{E}\|\nabla \ell_{T_i}(w_t)\|^2 + 2(1+\gamma L)^2(1+\gamma L)^2 \mathbb{E}_i b_i^2 \\ &\le 2(1+\gamma L)^4 (\|\nabla \ell_T(w_t)\|^2 + \sigma^2) + 2(1+\gamma L)^4 \tilde{b}_i \\ &\le 2(1+\gamma L)^4 (\frac{2}{C_1^2}\|\nabla \ell_T(w_t)\|^2 + 2\frac{C_2^2}{C_1^2} + \sigma^2) + 2(1+\gamma L)^4 \tilde{b}_i \\ &\le \frac{4(1+\gamma L)^4}{C_1^2}\|\nabla \ell_T(w_t)\| + \frac{4(1+\gamma L)^4 C_2^2}{C_1^2} + 2(1+\gamma L)^4(\sigma^2 + \tilde{b}), \end{aligned} \tag{39}$$

where constants $C_1, C_2 > 0$. $C_1 = 2 - (1+\gamma L)^2$ and $C_2 = ((1+\gamma L)^2 - 1)\sigma + (1+\gamma L)\gamma L b$.

□

Based on the above properties, we next characterize the convergence of one-step distilled meta-learning approach.

D.3 Convergence Results

Theorem D.5. *Let Assumption D.1 and Assumption D.2 hold, and apply Algorithm 1 to solve the objective function Equation (1). Choose the meta stepsize $\eta_t = 1/C_\eta \hat{L}(w_t)$ with $\hat{L}(w_t)$ given by Equation (35), where C_η>0 is a constant. For $\hat{L}(w_t)$ in Equation (35), we choose the batch size $B_k' = R(t)|\mathcal{D}_n|$ such that $|B_k'| \ge 2C_{\mathcal{L}}^2\sigma^2/(C_b b + (1+\gamma L)^2 L)^2$, here C_b and $C_{\mathcal{L}}$ are given by Equation (36). Define constants*

$$\begin{aligned} \varphi &= \frac{2-(1+\gamma L)^2}{C_{\mathcal{L}}}(1+\gamma L)^2 L + \frac{(2-(1+\gamma L)^2 C_b b)}{C_{\mathcal{L}}} + (1+\gamma L)^3 b, \\ \psi &= \frac{2-(1+\gamma L)^2}{C_{\mathcal{L}}}\left(\frac{1}{C_\eta} - \frac{1}{C_\eta^2}\left(\frac{A_w}{B}+1\right)\right) \\ \phi &= \frac{A_b}{LC_\eta^2}, \end{aligned} \tag{40}$$

where $C_b, C_{\mathcal{L}}, A_w, A_b$ are given by Equation (27) and Equation (36). Choose $\gamma < \sqrt{2}-1/L$, and choose C_η and B such that ψ>0. Then, Algorithm 1 attains a solution w_ζ such that

$$\mathbb{E}\|\nabla \mathcal{L}(w_\zeta)\| \le \frac{\Delta}{2\psi T} + \frac{\phi}{2\psi B} + \sqrt{\varphi(\frac{\Delta}{\psi T} + \frac{\phi}{\psi B}) + \varphi(\frac{\Delta}{2\psi T} + \frac{\phi}{2\psi B})^2} \tag{41}$$

The parameters φ, ψ and ϕ in Theorem D.5 take complicate forms.

Proof. Based on the smoothness of $\nabla\mathcal{L}(\cdot)$ established in Theorem D.3, we have

$$\mathcal{L}_{t+1} \le \mathcal{L}_t - \eta\left\langle \nabla\mathcal{L}(w_t), \frac{1}{B}\sum_{i\in\bar{B}_t}\right\rangle + \frac{L_{w_t}\eta^2}{2}\|\frac{1}{B}\sum_{i\in\bar{B}_t}\hat{G}_i(w_t)\|^2 \tag{42}$$

where $\bar{B}_t = R(t)B_t$ indicates the selected sample. Taking the conditional exception given w_t over the above inequality and noting that the randomness over η is independent of the randomness over $\hat{G}_i(w_t)$, we have

$$\begin{aligned}\mathbb{E}(\mathcal{L}(w_{t+1}) \mid w_t) \le \mathcal{L}(w_t) &- \frac{1}{C_\eta}\mathbb{E}(\frac{1}{L_{w_t}} \mid w_t)\|\nabla\mathcal{L}(w_t)\|^2 \\ &+ \frac{L_{w_t}}{2C_\eta^2}\mathbb{E}(\frac{1}{L_{w_t}^2} \mid w_t)\mathbb{E}(\|\frac{1}{B}\sum_{i\in\bar{B}_t}\hat{G}_i(w_t)\|^2 \mid w_t).\end{aligned} \tag{43}$$

Note that, conditioning on w_t,

$$\mathbb{E}\|\frac{1}{B}\sum_{i\in\bar{B}_t}\hat{G}_i(w_t)\|^2 \le \frac{1}{B}(A_w\|\nabla\mathcal{L}(w_t)\|^2 + A_b) + \|\nabla\mathcal{L}(w_t)\|^2 \tag{44}$$

where the inequality follows from Theorem D.4. Then, combining the Equation (54) and Equation (44), we have

$$\mathbb{E}(\mathcal{L}(w_{t+1}) \mid w_t) \le \mathcal{L}(w_t) - (\frac{1}{L_{w_t}C_\eta} - \frac{1}{L_{w_t}C_\eta^2}(\frac{A_w}{B}+1))\|\nabla\mathcal{L}(w_t)\|^2 + \frac{A_b}{L_{w_t}C_\eta^2 b}. \tag{45}$$

Recalling that $L_{w_t} = (1+\gamma L)^2 L + C_b b + C_{\mathcal{L}}C_\eta^2\mathbb{E}_i\|\nabla_{T_i}(w_t)\|$ and conditioning on w_t, we have $L_{w_t} \le L$ and

$$\begin{aligned}L_{w_t} &\le (1+\gamma L)^2 + C_b b + C_{\mathcal{L}}(\|\nabla\ell_t(w_t)\| + \sigma) \\ &\le (1+\gamma L)^2 + C_b b + C_{\mathcal{L}}(\frac{C_1}{C_2}+\sigma) + \frac{C_{\mathcal{L}}}{C_1}\|\nabla\mathcal{L}(w_t)\|,\end{aligned} \tag{46}$$

then, we have

$$\begin{aligned}\mathbb{E}(\mathcal{L}(w_{t+1}) \mid w_t) &\le \mathcal{L}(w_t)\frac{(\frac{1}{C_\eta} - \frac{1}{C_\eta^2}(\frac{A_w}{B}+1))\|\nabla\mathcal{L}(w_t)\|^2}{(1+\gamma L)^2 L + C_b b + C_{\mathcal{L}}(\frac{C_1}{C_2}+\sigma) + \frac{C_{\mathcal{L}}}{C_1}\|\nabla\mathcal{L}(w_t)\|} + \frac{1}{LC_\eta^2}\frac{A_b}{B} \\ &= \mathcal{L}(w_t)\frac{(\frac{1}{C_\eta} - \frac{1}{C_\eta^2}(\frac{A_w}{B}+1))\|\nabla\mathcal{L}(w_t)\|^2}{\frac{C_1}{C_{\mathcal{L}}}(1+\gamma L)^2 L + \frac{C_1 C_b b}{C_{\mathcal{L}}} + C_2 + C_1\sigma + \|\nabla\mathcal{L}(w_t)\|} + \frac{1}{LC_\eta^2}\frac{A_b}{B}\end{aligned} \tag{47}$$

Combining the definitions in Equation (40) with Equation (47) and taking the expectation over w_t, we have

$$\mathbb{E}\frac{\psi\|\nabla\mathcal{L}(w_t)\|^2}{\varphi + \|\nabla\mathcal{L}(w_t)\|} \le \mathbb{E}(\mathcal{L}(w_t) - \mathcal{L}(w_{t+1})) + \frac{\phi}{B}. \tag{48}$$

Telescoping the above bound over t from 0 to $T-1$ and choosing ζ from $0,\dots,T-1$ uniformly at random, we have

$$\mathbb{E}\frac{\psi\|\nabla\mathcal{L}(w_\zeta)\|^2}{\varphi + \|\nabla\mathcal{L}(w_\zeta)\|} \le \frac{\Delta}{T} + \frac{\phi}{B}. \tag{49}$$

Consider a function $f(x) = \frac{x^2}{c+x}, x{>}0$, where $c > 0$ is a constant. Simple computation shows that $f''(x) = \frac{2c^2}{(x+c)^3}{>}0$. Thus, applying Jensen's inequality, we have

$$\mathbb{E}\frac{\psi\|\nabla\mathcal{L}(w_\zeta)\|^2}{\varphi + \|\nabla\mathcal{L}(w_\zeta)\|} \le \frac{\Delta}{\psi T} + \frac{\phi}{\psi B}, \tag{50}$$

which further implies that

$$\mathbb{E}\|\nabla\mathcal{L}(w_\zeta)\| \leq \frac{\Delta}{2\psi T} + \frac{\phi}{2\psi B} + \sqrt{\varphi(\frac{\Delta}{\psi T} + \frac{\phi}{\psi B}) + (\frac{\Delta}{2\psi T} + \frac{\phi}{2\psi B})^2} \tag{51}$$

which finishes the proof. □

Theorem D.6. *Under the same setting of Theorem D.5, choose* $\gamma = \frac{1}{8L}, C_\eta = 80$. *We have*

$$\mathbb{E}\|\nabla\mathcal{L}(w_\zeta)\| \leq \mathcal{O}\left(\frac{1}{T} + \frac{\sigma^2}{B} + \sqrt{\frac{1}{T} + \frac{\sigma^2}{B}}\right) \tag{52}$$

Proof. Since $\gamma = \frac{1}{8L}$, we have $(1+\gamma L)^4 < e^{0.5} < 2$, and thus

$$\begin{aligned}
&A_w < 32, A_b < 8(\sigma + b)^2 + 4(\sigma^2 + \tilde{b}) \\
&C_{\mathcal{L}} < (\frac{5\rho}{32L} + \frac{\rho}{L}\frac{5}{16})\frac{5}{4} < \frac{5\rho}{8L}, \\
&C_{\mathcal{L}} > \frac{\rho}{L}(\gamma L) > \frac{\rho}{L}\gamma L > \frac{\rho}{16L}, \\
&C_b < \frac{15}{32}\frac{\rho}{L}\frac{1}{4} < \frac{\rho}{8L}
\end{aligned} \tag{53}$$

which, in conjunction with Equation (40), yields

$$\varphi \geq \frac{24L^2}{\rho} + \frac{37b}{16}, \psi \geq \frac{1}{80}\frac{4L}{5\rho}(1 - \frac{33}{80}) \geq \frac{L}{200}, \phi \geq \frac{2(\sigma+b)^2 + (\sigma^2 + \tilde{b})}{1600L}. \tag{54}$$

then, combining Equation (52) and Equation (54) yields

$$\mathbb{E}\frac{\psi\|\nabla\mathcal{L}(w_\zeta)\|^2}{\varphi + \|\nabla\mathcal{L}(w_\zeta)\|} \leq \frac{\Delta}{\psi T} + \frac{\phi}{\psi B}, \tag{55}$$

which further implies that

$$\begin{aligned}
\mathbb{E}\|\nabla\mathcal{L}(w_\zeta)\| &\leq \frac{\Delta}{2\psi T} + \frac{\phi}{2\psi B} + \sqrt{\varphi(\frac{\Delta}{\psi T} + \frac{\phi}{\psi B}) + (\frac{\Delta}{2\psi T} + \frac{\phi}{2\psi B})^2} \\
&\leq \mathcal{O}\left(\frac{1}{T} + \frac{\sigma^2}{B} + \sqrt{\frac{1}{T} + \frac{\sigma^2}{B}}\right)
\end{aligned} \tag{56}$$

□

Theorem 4.1 shows that the proposed method converges linearly with the number T of outer-loop meta iterations, and the convergence error decays linearly with the number B of sampled mini-batch. The convergence rate is further significantly affected by the number the inner-loop steps. Specifically, with respect to θ_S, meta-learning based converges exponentially fast as inner-loop cost increases.

E Empirical Results of High Noise Ratio

The experimental results of high noise rate are shown in A.

F More ablation studies

The ablation experiment mentioned in the main paper is supplemented in this session. In this section, we evaluate the effect of various modules on the CIFAR100 dataset with 40% symmetric noise. We first conduct the experiments on the CIFAR10 with symmetric noise use ResNet18 Backbone. And the other hyper-parameters are the same as those in the main paper.

Table A: Test accuracy on CIFAR10 / CIFAR100 with symmetric noise.

Dataset	Cifar10			Cifar100		
Method	M	$\tau = 0.8$	$\tau = 0.9$	M	$\tau = 0.8$	$\tau = 0.9$
F-Correction Patrini et al. (2017)	×	63.30	42.90	×	19.90	10.20
Co-teaching+ Yu et al. (2019)	×	67.40	47.90	×	27.90	13.70
DivideMix Li et al. (2020)	×	92.90	76.00	×	60.20	31.50
DM-AugDesc Nishi et al. (2021)	×	93.77	91.76	×	66.04	40.89
UNICON Karim et al. (2022)	×	93.90	90.80	×	63.90	44.80
Ours	×	**93.34**	**78.72**	×	**61.23**	**34.75**

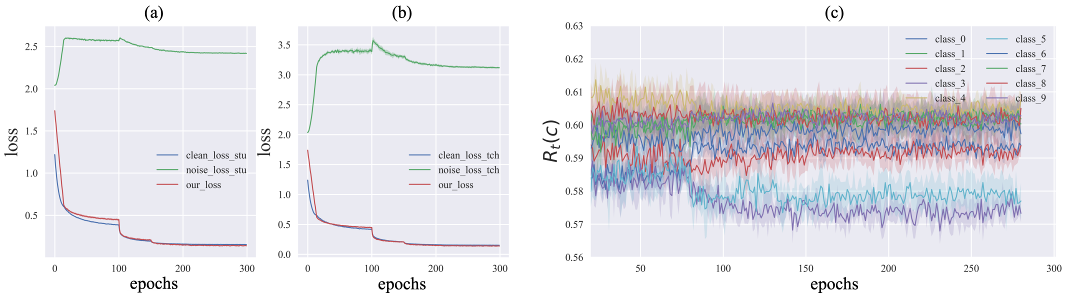

Figure 4: (a) and (b) indicate the loss curves of student and teacher network on CIFAR10 corrupted by 40% symmetric label noise, respectively. (c) indicates the curves of distillation criterion for each class, also known as $R_t(c)$.

F.1 Distillation Criterion

In this section, to illustrate the effectiveness of the meta-feedback mechanism, as shown in Figure 4 (a,b), by comparing our loss curves (Equation (2)) with clean and noisy cross-entropy curves. As training progresses, the meta-feedback mechanism makes the network to quickly fit clean samples and obtain curves consistent with clean CE losses. In addition, as shown in Figure 4 (c), we visualize the distillation criteria under each class.

F.2 The impact of Backbone

We use an 18-layer PreAct ResNet He et al. (2016) as the network backbone and train it using SGD with a batch size of 128. Other experimental settings follow the experiments in the main paper.

Table 9: Test accuracy on CIFAR10 with symmetric noise using ResNet18 Backbone.

Method	M	$\tau = 0.2$	$\tau = 0.5$	$\tau = 0.8$
Purify (Wu et al., 2021b)	×	93.4	90.3	69.9
DivideMix (Li et al., 2020)	×	96.1	94.6	92.9
C2D (Zheltonozhskii et al., 2022)	×	96.2	95.1	**94.3**
UNICON (Karim et al., 2022)	×	96.0	95.6	93.9
Ours	×	**96.3**	**95.8**	92.1

F.3 Estimated accuracy

As for performance measurements, we use the estimated accuracy in each mini-batch, *i.e.*, *estimated accuracy = (# of clean labels) / (# of all selected labels)*. Specifically, we sample $R(t)$ of small-loss instances in each mini-batch, and then calculate the ratio of clean labels in the small-loss instances. Intuitively, higher estimation accuracy means less noisy instances in the mini-batch after sample selection, and the method with higher estimation accuracy is also more robust to the label noise. Figure 5 shows the estimation accuracy of our method and Co-teaching for training data on the

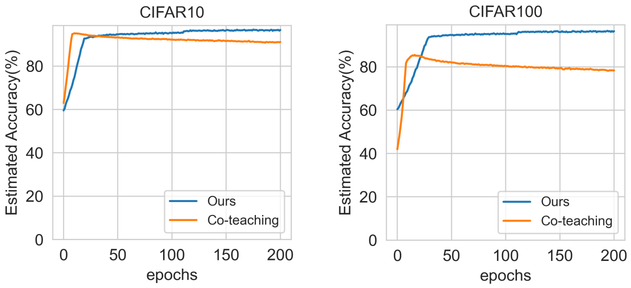

Figure 5: The Estimated Accuracy of the proposed method and Co-teaching.

CIFAR10/100 dataset respectively. This indicator shows the proportion of clean label samples to small loss samples. Our method can select more clean samples than Co-teaching along with the training.

G Instance-dependent Label Noise

We generate instance-dependent label noise following Xia et al. (2020). We compare our proposed method with the following representative approaches under instance-dependent noise: Peer Loss Liu & Guo (2020) introduces peer loss, a family of loss functions that enables training a classifier over noisy labels. CORES Cheng et al. (2021) introduces a sample sieve that is guaranteed to be robust to general instance-dependent label noise and sieve out corrupted examples. CAL Zhu et al. (2021) has proposed a second-order approach to transforming the challenging instance-dependent label noise into a class-dependent one. The initial learning rate is set to 0.05 for both teacher and student networks, and the detailed experimental results are shown in Table 4. The experimental results show that our proposed method is more robust and significantly outperforms the baseline methods in tackling instance-dependent label noise.

H Hyper-parameters

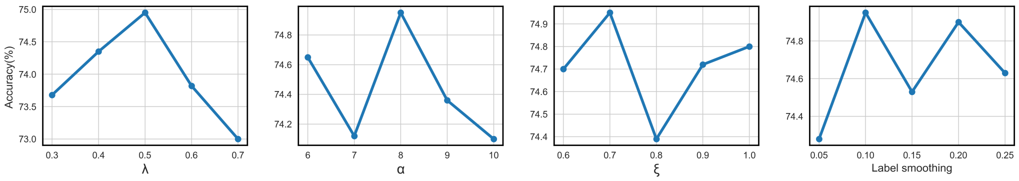

Figure 6: Accuracy with different hyper-parameters on CIFAR100 with 40% symmetric noise. The X-axis is the value of corresponding hyper-parameter and the Y-axis indicates accuracy.

In order to explore the influence of different hyper-parameters on the experimental results, we evaluate the impact of four hyper-parameters on the CIFAR100 dataset with 40% symmetric noise and plot the accuracy curve for four hyper-parameters under different values. We use the same hyper-parameters with $\lambda = 0.5$, $\alpha = 8.0$, $\xi = 0.7$, and label smoothing = 0.10. As shown in Figure 6 and Figure 7 , λ is set from 0.3 to 0.7, $\lambda = 0.5$ results in a better performance. We set $\alpha_{min} = 6.0$, as α increases, $\alpha = 8.0$ results in a better performance. We pick $\xi_{min} = 0.6$, $\xi_{max} = 1.0$, $\xi = 0.7$ outperforms than others. Label smoothing is set from 0.05 to 0.25, label smoothing = 0.10 results in a better performance. The biggest change by sweeping these hyper-parameters is 2% which shows that our proposed method is insensitive to the hyper-parameters.

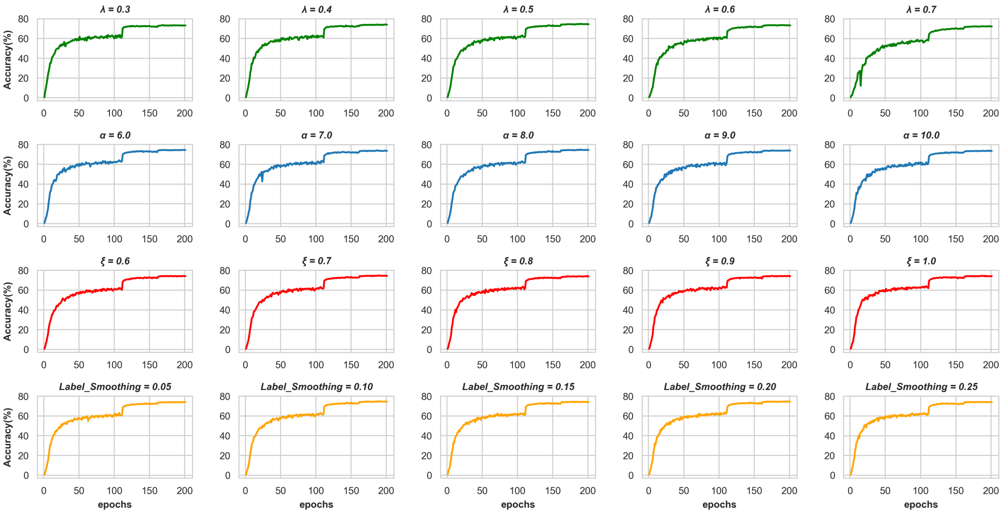

Figure 7: Accuracy on CIFAR100 with 40% symmetric noise of different hyper-parameters.