Variational Rectification Inference for Learning with Noisy Labels

Haoliang Sun [1†], Qi Wei [2†], Lei Feng [2], Yupeng Hu [1*], Fan Liu [3], Hehe Fan [4], Yilong Yin [1]

[1]School of Software, Shandong University, Jinan, China.
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore.
[3]School of Computing, National University of Singapore, Singapore.
[4]School of Computer Science and Technology, Zhejiang University, Hangzhou, China.

*Corresponding author(s). E-mail(s): huyupeng@sdu.edu.cn;
Contributing authors: haolsun@sdu.edu.cn; 1998v7@gmail.com; ylyin@sdu.edu.cn; feng0093@e.ntu.edu.sg; liufancs@gmail.com; hehefan@zju.edu.cn;
†Equal contribution

Abstract

Label noise has been broadly observed in real-world datasets. To mitigate the negative impact of overfitting to label noise for deep models, effective strategies (*e.g.*, re-weighting or loss rectification) have been broadly applied in prevailing approaches, which have been generally learned under the meta-learning scenario. Despite robustness of noise achieved by the probabilistic meta-learning models, they usually suffer from model collapse that degenerates generalization performance. In this paper, we propose variational rectification inference (VRI) to formulate the adaptive rectification for loss functions as an amortized variational inference problem and derive the evidence lower bound under the meta-learning framework. Specifically, VRI is constructed as a hierarchical Bayes by treating the rectifying vector as a latent variable, which can rectify the loss of the noisy sample with the extra randomness regularization and be therefore more robust to label noise. To achieve the inference of the rectifying vector, we approximate its conditional posterior with an amortization meta-network. By introducing the variational term in VRI, the conditional posterior is estimated accurately and avoids collapsing to a Dirac delta function, which can significantly improve the generalization performance. Given a set of clean meta-data, VRI can be efficiently meta-learned within the bi-level optimization programming. Besides, theoretical analysis guarantees that the meta-network can be efficiently learned with our algorithm. Extensive comparison experiments and analyses demonstrate its effectiveness for robust learning with noisy labels.

Keywords: Learning with Noisy Labels, Meta-learning, Variational Inference, Loss Correction.

1 Introduction

Learning from noisy labels (Xia et al, 2023; Yuan et al, 2023; Huang et al, 2023; Wei et al, 2023; Xu et al, 2021a; Ortego et al, 2021; Gudovskiy et al, 2021) poses great challenges for training deep models, whose performance heavily relies on large-scaled labeled datasets. Annotating training data with high confidence would be resource-intensive, especially for some domains with ambiguous labels, such as medical image segmentation and multi-modal learning (Pu et al, 2023; Liu et al, 2023). In this case, label noise would inevitably arise since there is usually a lack of experts for accurate annotation (Ge et al, 2023).

Re-weighting (Kumar et al, 2010; Zadrozny, 2004; Jiang et al, 2018; Shu et al, 2023) and loss rectification (Zhang et al, 2021a; Vahdat, 2017; Yao et al, 2020; Sun et al, 2022) are two effective strategies to reduce the bias of learning caused by noisy labels. The basic idea is to construct a weight function or transition matrix to mitigate the effect of noisy samples. Although those strategies have been broadly applied, there are two limitations. 1) The form of the weighting functions needs to be manually specified under certain assumptions on the data distribution, restricting its expandability in the real world (Shu et al, 2019). 2) Hyper-parameters in these functions are usually tuned by cross-validation, which suffers from the issue of scalability (Franceschi et al, 2018).

A family of approaches based on meta-learning have been recently proposed for noisy labels (Shu et al, 2023; Xu et al, 2021a; Zheng et al, 2021; Zhang et al, 2019; Shu et al, 2019; Zhao et al, 2023; Sun et al, 2022; Wu et al, 2021). By introducing a small meta-data set with completely clean labels, an effective weighting (*e.g.*, meta-weight-net (Shu et al, 2019)) or correction (*e.g.*, meta label correcter (Zheng et al, 2021)) function can be meta-learned under the meta-learning scenario, omitting the prior assumption for these functions and avoiding manually tuning of hyper-parameters (Ren et al, 2018). To enhance the interpretability and generalization ability, Bayesian meta-learning (Zhao et al, 2023; Sun et al, 2022) has been applied to model the uncertainty of parameters and achieved a favorable performance for learning with noisy labels. The probabilistic meta-weight-net (Zhao et al, 2023) applies a Bayesian weight network to estimate the distribution of the sample weight. The probabilistic formulation is elegant. However, the weighting network merely takes the loss as the input to compute the sample weight, it would be deficient in controlling the learning process and result in low expression capability (Sun et al, 2022). To strengthen the capability of the meta-network, a rectification network has been proposed in (Sun et al, 2022) to achieve rectifying the training process with an estimated vector. By treating the rectifying vector as a latent variable, the predictive posterior can be estimated by Monte-Carlo (MC) approximation. Although the MC approximation has achieved desirable effectiveness for rectifying the bias of the learning process, we have observed that there would exist model collapse (Iakovleva et al, 2020) where the conditional prior collapses to a Dirac delta function and the model degenerates to a deterministic parameter generating network, especially for a small sampling number in MC. This collapse may degrade the generalization performance of the model, which is illustrated in Fig. 1.

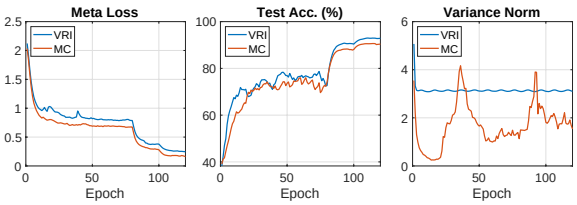


Fig. 1 Illustration of model collapse with 70% uniform noise. There exists a gap between MC and VRI in the meta-loss curve. The norm of variance for the rectification vector of MC degenerates into zero in some cases. The generalization performance is also degraded.

In this work, to tackle the model collapse issue in MC approximation, we propose to formulate learning rectification process as an amortized variational inference problem and derive the evidence lower bound (ELBO) under the meta-learning framework. We construct variational rectification inference (VRI) to achieve an adaptively rectifying learning process for noisy labels as shown in Fig. 2. We treat the rectifying vector as a latent variable and build a hierarchical Bayes under the setting of the meta-learning scenario. We introduce an amortization meta-network to estimate the posterior distribution of the rectifying vector and achieve a rectified prediction via Monte

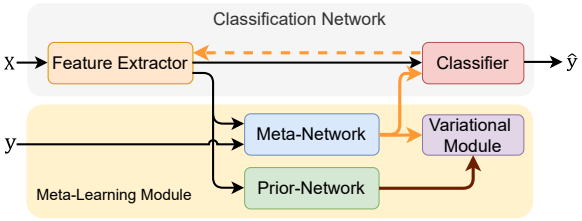


Fig. 2 The meta-network can generate the rectifying vector to integrate into the inference of the classification network. The variational module can avoid model collapse via a prior network.

Carlo sampling. The proposed meta-network is built to leverage the feature embedding and corresponding label as inputs, which can faithfully exploit sufficient information lying in the feature space and significantly improve the generalization performance of the classification network.

By building a variational term with a prior network to constraint the posterior, VRI can avoid the model collapse in MC approximation with limited samples and further enhance the capability of inference for the unbiased estimation of the predictive posterior. VRI can be integrated into the meta-learning framework to achieve adaptive rectification for noisy samples. By introducing the meta-data, we conduct the meta-learning process with a bi-level programming schema and achieve robust learning with label noise.

Our contributions can be summarized in four aspects.

- We formulate the learning rectification process as an amortized variational inference problem and derive the ELBO under the meta-learning framework.
- We build a variational constraint for the posterior, which can avoid the model collapse in MC approximation.
- We propose the learning framework of VRI, which can be efficiently solved via bi-level optimization, exhibiting virtues of robust learning for label noise.
- We provide the rigorously theoretical guarantee for the convergence of the proposed algorithm for the meta-network.

We conduct extensive experiments on five challenging benchmark datasets under a variant of noise types. Our VRI outperforms the state-of-the-art in most cases. Additional promising results and further complementary analysis also demonstrate the effectiveness of VRI.

The rest of this paper is organized as follows. Section 2 introduces related works and discusses the relations to our work. Section 3 includes the problem setting, preliminaries and our noise-robust method Variational Rectification Inference (VRI). We also provide the theoretical provide for VRI. Section 4 reports experimental results on three noise types and various datasets. We test the performance of VRI on the restricted scenario (*i.e.* training without the meta-set). Finally, Section 5 gives a conclusion.

2 Related Work

Re-weighting. The main idea of the sample re-weighting strategy is to assign a small weight to samples with corrupted labels (Shu et al, 2019, 2023). Since the clean example usually have a small loss and deep models can memorize them at the beginning of the training steps (Arpit et al, 2017), samples with the lower loss are selected for learning at each epoch in (Shen and Sanghavi, 2019; Cui et al, 2019). Based on this assumption, MentorNet (Jiang et al, 2018) adopts the idea of curriculum learning to train a mentor network to guide learning of the student classification network. A Bayesian model (Wang et al, 2017) has also been extended to inferring the latent variables of sample weights for handling label noise. To avoid manually designing or tuning weighting functions, meta-learning has been introduced to learn to generate weights from a meta-data set with clean labels. The pioneering work, inspired by the two nested loops of optimization (Finn et al, 2017), sets the weight value as trainable parameters (Ren et al, 2018) and achieves a dynamically weighting strategy. Meta-Weight-Net (Shu et al, 2019) further improves the scalability of the weighting space by directly generating weights via an MLP and being learned under the meta-learning scenario.

Correcting. There are plenty of methods working on loss or label correction of the objective function, which can be essentially categorized into three aspects. 1) A confusion matrix (Sukhbaatar et al, 2015; Han et al, 2018a; Tanno et al, 2019; Yao et al, 2020), restoring the transition probability between the true label and the noisy one, is

estimated and multiplied to the prediction vector. This can be considered as a smooth regularization for the prediction to mitigate the impact of corrupted labels. The following works (Hendrycks et al, 2018; Pereyra et al, 2017) introduce a set of clean anchor-data to improve the estimation accuracy of the confusion matrix. Recently, an MC approximation framework (Sun et al, 2022) is proposed to learn to generate the rectification vector for loss functions, demonstrating the superiority of handling the sample ambiguity in noisy data. 2) Another family of methods, such as Reed (Reed et al, 2015), D2L (Ma et al, 2018), S-Model (Goldberger and Ben-Reuven, 2017), includes extra inference steps to correct corrupted labels for the following optimization. By leveraging clean meta-data, MSLC (Wu et al, 2021; Zheng et al, 2021) learns an efficient label corrector to reduce label noise. 3) Designing appropriate loss functions also provide an effective solution to significantly enhance the robustness of deep models. Noise-tolerant losses, such as mean absolute error (MAE), have been theoretically analyzed for noisy labels in (Ghosh et al, 2017). The following works (Zhang and Sabuncu, 2018; Wang et al, 2019) further improve the performance of MAE on challenging datasets with generalized MAE and cross-entropy losses. Recently, a dynamically weighted bootstrapping loss (Arazo et al, 2019) has been designed for noisy samples based on an unsupervised beta mixture model.

Meta-learning, leverages shared knowledge among a series of tasks to improve the performance of the current task, which has made great breakthroughs recently (Hospedales et al, 2022). The typical idea is to parameterize a trainable function as the meta-learner to generate the parameters or statistics for base learners, which can be regarded as the "black-box" adaptation. By introducing the clean meta-data set, the aforementioned strategies (*e.g.*, re-weighing (Ren et al, 2018; Zhao et al, 2023; Shu et al, 2019, 2023) or loss correction (Zhang et al, 2019; Wu et al, 2021; Zheng et al, 2021; Sun et al, 2022)) can be meta-learn in a data-driven way, avoiding manually tuning hyper-parameters with the validation set in conventional methods (Ren et al, 2018).

Semi-supervised learning (SSL), builds a labeled set that contains confident examples by sample selection strategies and employs modern SSL techniques (e.g., FixMatch (Sohn et al, 2020) and MixMatch (Berthelot et al, 2019)) to effectively leverage the labeled set and the remaining unlabeled set (Li et al, 2020; Liu et al, 2020; Wei et al, 2020; Xia et al, 2023). Compared with other branchs, SSL-based methods have achieved state-of-the-art performance on image benchmarks since they can incorporate prior knowledge to exploit discriminative information from finite training samples. *However, the data generative process has the impact on the performance of SSL methods* (Yao et al, 2023). *When the image feature is the cause of the label, the performance of SSL methods is worse than model-based methods, e.g., the method based on the confusion matrix* (Yao et al, 2020).

Other methods. Additional lines of methods for handling label noise include 1) data augmentation (Zhang et al, 2018; Nishi et al, 2021), exploring different augmentation policy to mitigate the side-effect of noisy labels, 2) sample selection (Han et al, 2018b; Yu et al, 2019; Song et al, 2019; Wei et al, 2022; Xia et al, 2023), designing an effective selection strategy to select clean data from the noisy training set, 3) early-learning regularization (Liu et al, 2020), combating noisy signal by regularizing model in the early learning stage, 4) contrastive learning (Wei et al, 2023; Li et al, 2022), combating noisy signal via enhancing representation ability of deep models.

Relations to us. In contrast to prevailing works, we formulate the rectification process as an amortized variational inference problem. By building a hierarchical Bayes model, VRI exhibits the favorable property of handling the sample ambiguity. The variational term in VRI can avoid the model collapse exiting in those MC approximation methods.

3 Variational Rectification Inference

We propose variational rectification inference (VRI) for adaptively rectifying the learning processing under the setting of meta-learning, which effectively mitigates the side-effect of noisy labels. VRI includes a meta-network that generates a rectifying vector to support the learning of the classification network. The whole learning procedure is formulated as an amortized variational

inference problem. We integrate VRI into the bi-level optimization steps and achieve meta-learning the rectifying process.

3.1 Preliminaries

Robust Learning with Meta-Data. Given the training set $\mathcal{D}_N = \{\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\}_{i=1}^N$ with noisy labels, the aim for robust learning is to achieve good generalization performance on the clean testing set. Under the setting of meta-learning, we construct a set of clean examples $\mathcal{D}_M = \{\tilde{\mathbf{x}}^{(i)}, \tilde{\mathbf{y}}^{(i)}\}_{i=1}^M$, regarded as the meta-data set, which is smaller than the training set $\mathcal{D}_N$ of $N \gg M$. We usually choose the validation set as the meta-data set in practice. Therefore, the meta-learning process can be considered as learning to tune the hyper-parameters in a data-driven way.

Rectification for the loss function. Loss rectification (Hendrycks et al, 2018; Sun et al, 2022) is an effective tool for mitigating the effect of the label noise with meta-data. There are essentially two networks in the learning framework. The meta-network $V(\mathbf{y}^{(i)}, \mathbf{z}^{(i)}; \phi)$ with the parameter of ϕ is trained with the meta-data set to take the feature embedding $\mathbf{z}^{(i)}$ and label $\mathbf{y}^{(i)}$ of the example $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$ as input and generates a vector $\mathbf{v}^{(i)}$ to rectify the learning process of the classification network. Let $\odot$ denote the element-wise product. By multiplying $\mathbf{v}^{(i)}$ on the logits calculated from the classification network $\mathbf{v}^{(i)} \odot F(\mathbf{x}^{(i)}; \theta)$, the rectified loss with noisy labels can still produce effective update direction. Therefore, the negative impact from corrupted labels in the noisy training set can be mitigated.

3.2 Variational Rectification Inference

The inference process in our framework is built as a hierarchical Bayes model. From the probabilistic perspective, we treat the rectifying vector as the latent variable and compute the posterior distribution $p(\mathbf{v}|\mathbf{x}, \mathbf{y})$ given the observation of the sample. Our goal of this task is to accurately approximate the conditional predictive distribution with parameters θ by maximizing its log-likelihood

$$\max \log p_\theta(\mathbf{y}|\mathbf{x}) = \log \int p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{v}) p(\mathbf{v}|\mathbf{x}) \mathrm{d}\mathbf{v}. \tag{1}$$

The rectified learning process in this work consists of two steps. First, form the posterior distribution $p(\mathbf{v}|\mathbf{x})$ over $\mathbf{v}$ for each sample $(\mathbf{x}, \mathbf{y})$. Then, calculate the posterior predictive $p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{v})$. Since inferring the posterior $p(\mathbf{v}|\mathbf{x})$ is generally intractable, we resort to approximating it by leveraging a variational distribution $q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y})$. We minimize the Kullback–Leibler (KL) divergence D_{KL} between $q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y})$ and $p(\mathbf{v}|\mathbf{x}, \mathbf{y})$ to obtain the variational distribution

$$\min D_{\mathrm{KL}}[q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y})||p_\theta(\mathbf{v}|\mathbf{x}, \mathbf{y})]. \tag{2}$$

We can then derive the tractable evidence lower bound (ELBO) of the conditional predictive distribution to approximate the posterior $p(\mathbf{v}|\mathbf{x}, \mathbf{y})$ by applying the Bayes' rule

$$\begin{aligned} &\max \log p_\theta(\mathbf{y}|\mathbf{x}) \geq \mathcal{L}_{\mathrm{ELBO}} \\ &= \mathop{\mathbb{E}}_{q_\phi(\mathbf{v}|\mathbf{x},\mathbf{y})} p_\theta(\mathbf{y}|\mathbf{x}, \mathbf{v}) - D_{\mathrm{KL}}[q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y})||p_\omega(\mathbf{v}|\mathbf{x})]. \end{aligned} \tag{3}$$

The first term of the ELBO is the predictive log-likelihood conditioned on the input $\mathbf{x}$ and the inferred rectifying vector $\mathbf{v}$. Maximizing it can achieve accurate rectified prediction for each sample. The second term is to minimize the discrepancy between the variational distribution $q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y})$ and the prior $p_\omega(\mathbf{v}|\mathbf{x})$ assigned to a certain distribution form. The detailed derivation of the ELBO is provided in Appendix A.1. Once we obtain $q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y})$, the inference procedure can be summarized as 1) forming the variational distribution $q_\phi(\cdot)$ on the fly with amortized variational inference (AVI); 2) calculating the posterior predictive distribution $p(\mathbf{y}|\mathbf{x}, \mathbf{v})$ via Monte Carlo estimation.

Application Details. In practice, we assume that the latent variable $\mathbf{v}$ obeys the factorized Gaussian distribution $\mathbf{v} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. There are three networks in our framework. The classification network F with parameters θ works on the basic categorizing task. We implement the variational distribution with an amortization meta-network V with parameters ϕ that takes a pair of the feature embedding and label of the sample as input and outputs the parameters $(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$ of a factorized Gaussian distribution q. By sampling a vector $\mathbf{v}^{(i)}$ from q, F can compute a rectified prediction $\hat{\mathbf{y}}^{(i)}$. The prior is also implemented as a network H with parameters ω that takes the

feature as inputs and outputs another factorized Gaussian distribution p. To enable an unbiased estimate of the objective in Eq. (1), we adopt the Monte Carlo Sampling strategy that repeats the above process multiple times and averages all predictions. Note that it is commonly intractable to back-propagate through sampling operations, we solve it by applying the reparameterization trick proposed in (Kingma and Welling, 2014) as

$$\mathbf{v} = \boldsymbol{\mu} + \boldsymbol{\sigma} \cdot \boldsymbol{\epsilon} \quad \text{with} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathrm{I}). \tag{4}$$

We denote $\mathrm{RP}(\cdot)$ as the sampling operation with the reparameterization trick for simplicity in the following section.

3.3 Meta-Learning Process

We present the practical objective function to achieve jointly learning the three networks of $F_\theta(\cdot)$, $V_\phi(\cdot)$, and $H_\omega(\cdot)$. By formulating the problem as a meta-learning task, we conduct bi-level optimization programming to solve it. The exhaustive derivation for each updating step is also provided in the following.

3.3.1 The practical objectives

We derive the practical objective from the ELBO in Eq. (3). To improve the generalization performance on noisy labels, the empirical loss for our prediction model $F(\cdot)$ of N samples is rectified with the support of the meta-network

$$\mathcal{L}^{emp}(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(\mathbf{y}^{(i)}, \mathbf{v}^{(i)} \odot F_\theta(\mathbf{x}^{(i)})), \tag{5}$$

where $\mathbf{v}^{(i)}$ is a rectifying vector sampled from the variational posterior $q^{(i)}(\mathbf{v})$ with the form of the factorized Gaussian $\mathcal{N}(\boldsymbol{\mu}^{(i)}, {\boldsymbol{\sigma}^{(i)}}^2)$, whose parameters are generated by the amortization meta-network $(\boldsymbol{\mu}^{(i)}, \boldsymbol{\sigma}^{(i)}) \leftarrow V_\phi(F'_{\theta'}(\mathbf{x}^{(i)}), \mathbf{y}^{(i)})$. $F'_{\theta'}$ is the feature extractor in F_θ, where $\theta' \subset \theta$. To stabilize the learning process, we bound $\mathbf{v}^{(i)}$ with the sigmoid function. The form of the loss function $L(.)$ is flexible, we adopt the basic cross-entropy loss with the softmax function.

For the objective *w.r.t.* F_θ, the aim is to achieve the unbiased estimation of the conditional predictive distribution, which can be attained with Monte Carlo sampling. Recall reparameterization

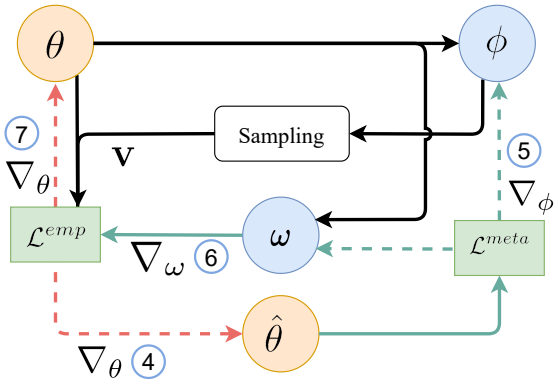


Fig. 3 Flowchart of the learning algorithm. The solid and dashed lines denote forward and backward propagation, respectively. For each iteration, the meta-network ϕ generates the distribution of $\mathbf{v}$, and then produces multiple examples via the sampling module to estimate the predictive distribution. By computing the gradient through the update step 4, the meta-network can be trained in step 5. The prior network is also jointly optimized in step 6. The classification network θ will be updated with support of the learned meta-network in step 7.

(RP) in Eq. (4), supposing the sampling number for $\mathbf{v}$ is k, the ultimate objective for the ELBO in Eq. (3) can be written as

$$\begin{aligned}
&\arg\min_\theta \mathcal{L}^{emp}(\theta) = \\
&\frac{1}{kN} \sum_{i=1}^{N} \sum_{j=1}^{k} L(\mathbf{y}^{(i)}, \mathrm{RP}^{(j)}[V(F'_{\theta'}(\mathbf{x}^{(i)}), \mathbf{y}^{(i)})] \odot F_\theta(\mathbf{x}^{(i)})) \\
&+ \lambda D_{\mathrm{KL}}[V(F'_{\theta'}(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}) || H(F'_{\theta'}(\mathbf{x}^{(i)}))].
\end{aligned} \tag{6}$$

Here, we add a coefficient λ to weight the KL term as beta-VAE (Higgins et al, 2017). The KL term can be considered as a regularizer to the meta-network, which is proved to improve the stability of the meta-learning process as indicated in (Bao et al, 2021).

The Monte Carlo estimation strategy for the predictive distribution ensures an efficient feed-forward propagation phase of the model during training. We further analyze the effect of the sampling number in the experimental section.

For the meta objective *w.r.t.* V_ϕ, the performance of the meta-network is evaluated on the meta-data set $\mathcal{D}_M$. Since the feed-forward propagation in Eq. (6) involves the support of V_ϕ and H_ω, we denote the updated θ as $\theta^*(\phi, \omega)$. Therefore, the objective for the meta-network with

meta-data $(\tilde{\mathbf{x}}^{(i)}, \tilde{\mathbf{y}}^{(i)})$ can be written as

$$\arg\min_{\phi,\omega} \mathcal{L}^{meta}(\phi,\omega) = \frac{1}{M}\sum_{i=1}^{M} L(\tilde{\mathbf{y}}^{(i)}, F(\tilde{\mathbf{x}}^{(i)}; \theta^*(\phi,\omega))). \tag{7}$$

By minimizing Eq. (7) *w.r.t.* ϕ involved in the updated F_{θ^*}, the learned V_{ϕ^*} can achieve unbiased estimation for the posterior and generate rectifying vectors with high fidelity to guide following updates of θ. Also, the prior network H_{ω^*} restricts V_{ϕ^*} to avoid collapsing to produce Dirac delta functions.

3.3.2 Bi-level optimization

We build an iterative optimization algorithm within the bi-level programming framework (Franceschi et al, 2018) to obtain the optimal parameters $\{\theta^*, \phi^*, \omega^*\}$ as follows

$$\begin{aligned}\phi^*, \omega^* &= \arg\min_{\phi,\omega} \mathcal{L}^{meta}(\theta^*(\phi,\omega,\mathcal{D}_N), \mathcal{D}_M), \text{ s.t.} \\ \theta^*(\phi,\omega,\mathcal{D}_N) &= \arg\min_{\theta} \mathcal{L}^{emp}(\phi,\omega,\theta,\mathcal{D}_N).\end{aligned} \tag{8}$$

We adopt stochastic gradient descent (SGD) to solve (8). Since the prediction from F_θ is rectified by V_ϕ, the gradient for θ is closely related to ϕ and ω. Thus, $\hat{\theta}(\phi,\omega)$ denotes that the updated $\hat{\theta}$ is the function of ϕ and ω. Here, we assign a learning rate of α. By sampling a mini-batch of n training examples $\{(\mathbf{x}^i, \mathbf{y}^i)\}_{i=1}^n$, the updating step of the classification network F_θ *w.r.t.* Eq. (6) can be written as

$$\begin{aligned}\hat{\theta}^{(t)}(\phi,\omega) = \theta^{(t)} - \alpha\nabla_\theta \widetilde{\mathcal{L}}^{emp}(\theta), \quad \text{where} \quad \widetilde{\mathcal{L}}^{emp}(\theta) = \\ \frac{1}{kn}\sum_{i=1}^{n}\sum_{j=1}^{k} L(\mathbf{y}^{(i)}, \mathrm{RP}^{(j)}[V(F'_{\theta'^{(t)}}(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}; \phi)] \odot F_{\theta^{(t)}}(\mathbf{x}^{(i)})) \\ + \lambda D_{\mathrm{KL}}[V(F'_{\theta'^{(t)}}(\mathbf{x}^{(i)}), \mathbf{y}^{(i)}; \phi) || H(F'_{\theta'^{(t)}}(\mathbf{x}^{(i)}; \omega)].\end{aligned} \tag{9}$$

Given a mini-batch of m meta samples $\{(\tilde{\mathbf{x}}^i, \tilde{\mathbf{y}}^i)\}_{i=1}^m$, the learning of ϕ and ω can be achieved by back-propagating through the learning process of θ. Specifically, after obtaining $\hat{\theta}^{(t)}(\phi,\omega)$ with fixed ϕ and ω in Eq. (9), the parameter of ϕ in the meta-network $V_\phi(\cdot)$ can be updated *w.r.t.* the objective in Eq. (7)

$$\phi^{(t+1)} = \phi^{(t)} - \eta\frac{1}{m}\sum_{i=1}^{m}\nabla_\phi L(\tilde{\mathbf{y}}^{(i)}, F(\tilde{\mathbf{x}}^{(i)}; \hat{\theta}^{(t)}(\phi,\omega))), \tag{10}$$

Algorithm 1 The Bi-level optimization for VRI

Require: Training set $\mathcal{D}_N$, meta set $\mathcal{D}_M$, batch size n, m, outer iterations T, step size α, η, sampling number k,
Ensure: Optimal θ^*

```
Initialize parameters θ^(0), φ^(0), and ω^(0)
for t ∈ {1, ..., T} do
    SampleBatch(D_N, n), SampleBatch(D_M, m)
    Form learning process of θ̂^(t)(φ, ω)  ▷ Eq. (9)
    Optimize φ^(t) with θ̂^(t)(φ)          ▷ Eq. (10)
    Optimize ω^(t) with θ̂^(t)(ω)          ▷ Eq. (11)
    Optimize θ^(t) using the updated φ^(t+1)
                                          ▷ Eq. (12)
end for
```

where η denotes the step size. Similar update steps for the prior network can be written as

$$\omega^{(t+1)} = \omega^{(t)} - \eta\frac{1}{m}\sum_{i=1}^{m}\nabla_\omega L(\tilde{\mathbf{y}}^{(i)}, F(\tilde{\mathbf{x}}^{(i)}; \hat{\theta}^{(t)}(\phi,\omega))) \tag{11}$$

This bi-level programming manner results in the best hypothesis on the meta-data set, whose theoretical guarantee has been rigorously studied in (Bao et al, 2021).

Once V_ϕ has been updated, we utilize the current training batch to conduct robustly learning of the classification network $F_{\theta^{(t)}}$

$$\begin{aligned}\theta^{(t+1)} = \theta^{(t)} - \alpha\frac{1}{kn}\sum_{i=1}^{n}\sum_{j=1}^{k}\nabla_\theta L(\mathbf{y}^{(i)}, F(\mathbf{x}^{(i)}; \theta^{(t)}) \\ \odot \mathrm{RP}^{(j)}[V(F'(\mathbf{x}^{(i)}; \theta'^{(t)}), \mathbf{y}^{(i)}; \phi^{(t+1)})]).\end{aligned} \tag{12}$$

We summarize the overall updating steps in Algorithm 1 and illustrate the main information flow in Fig. 3. Estimating the conditional predictive distribution can be efficiently implemented via the Monte Carlo sampling of averaging k results. Indeed, by introducing the variational term, VRI merely require a small number (*e. g.*, $k = 1$ or 2) of samples for the good performance. By applying RP trick, the sampling operation is tractable for gradient computation. Therefore, all gradients, including the bi-level programming process, can be efficiently calculated by prevailing differentiation tools.

3.4 Convergence Analysis

The convergence of our proposed Algorithm 1 can be rigorously theoretically guaranteed. Since the meta-network $V(\phi)$ is crucial in our framework, we prove that the algorithm for $V(\phi)$ can converge to the stationary point of the meta loss function under some mild conditions. To facilitate the proof, we adopt the stochastic gradient $\nabla\widetilde{\mathcal{L}}^{meta}\left(\hat{\theta}^{(t)}\left(\phi^{(t)}\right)\right)$ in the following, which is identical to uniformly drawing a mini-batch of samples at random in Eq. (9).

Lemma 1 (Smoothness). *Suppose the loss function L w.r.t. θ in Eq. (7) is ℓ-smooth and τ-Lipschitz, the KL term D_{KL} w.r.t. the output of $V(\phi)$ has the o-bounded gradient, and $V(\phi)$ is differential with the δ-bounded gradient and twice differential with its ζ-bounded Hessian. Then the meta loss function w.r.t. θ is $\hat{\ell}$-smooth.*

Proof. See Appendix A.2. □

Lemma 1 implies that the meta loss *w.r.t.* the meta-network is smooth-bounded. We provide the convergence rate in Theorem 1 with the support of this essential property.

Theorem 1 (Convergence Rate). *Assume that the variance of the stochastic gradient $\nabla\widetilde{\mathcal{L}}^{meta}\left(\hat{\theta}^{(t)}\left(\phi^{(t)}\right)\right)$ is bounded $\mathbb{E}\left[\left\|\nabla\widetilde{\mathcal{L}}^{meta}\left(\hat{\theta}^{(t)}\left(\phi^{(t)}\right)\right)-\nabla\mathcal{L}^{meta}\left(\hat{\theta}^{(t)}\left(\phi^{(t)}\right)\right)\right\|_2^2\right] \leq \sigma^2 < \infty$. Following directly from Lemma 1, let the learning rate α_t satisfies $\alpha_t = \min\{1, \frac{\kappa}{T}\}$, for some $\kappa > 0$, such that $\frac{\kappa}{T} < 1$, and $\eta_t, 1 \leq t \leq T$ is a monotone descent sequence, $\eta_t = \min\{\frac{1}{\hat{\ell}}, \frac{C}{\sigma\sqrt{T}}\}$ for some $C > 0$, such that $\frac{\sigma\sqrt{T}}{C} \geq \hat{\ell}$ and $\sum_{t=1}^{\infty}\eta_t \leq \infty, \sum_{t=1}^{\infty}\eta_t^2 \leq \infty$. Then we have*

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\left\|\nabla\mathcal{L}^{meta}\left(\hat{\theta}^{(t)}(\phi^{(t)})\right)\right\|_2^2\right] \leq \mathcal{O}(\frac{1}{\sqrt{T}}). \tag{13}$$

Proof. See Appendix A.2. □

More specifically, Theorem 1 implies that our learning algorithm VRI can achieve $\mathbb{E}\left[\left\|\nabla\mathcal{L}^{meta}\left(\hat{\theta}^{(t)}\left(\phi^{(t)}\right)\right)\right\|_2^2\right] \leq \epsilon$ in $\mathcal{O}(1/\epsilon^2)$ steps. As the iteration step increases, the algorithm would ultimately converge to a stationary point.

4 Experiments

We conduct classification experiments with variant noise types on five benchmarks, including three real-world datasets, and obtain better performance compared with the state-of-the-art (SOTA) method. Exhaustive analysis further demonstrates the virtue of the proposed model on LNL task. The code is now available at https://github.com/haolsun/VRI.

4.1 Setup

Datasets. We evaluate VRI on five benchmarks of CIFAR-10, CIFAR-100, Clothing1M (Xiao et al, 2015), and Food-101N (Lee et al, 2018), and follow the consistent experimental protocol in (Shu et al, 2019; Zhang et al, 2021b) for the fair comparison. We randomly select 1000 training samples (2%) as meta data for CIFAR-10 & 100. For Clothing1M and Food-101N, we use the validation set for meta-learning. More details for constructing those datasets are provided as following.

CIFAR-10 (Krizhevsky et al, 2009) dataset consists of 60,000 images of 10 categories. We adopt the splitting strategy in (Shu et al, 2019) by randomly selecting 1,000 samples from the training set to construct the meta dataset. We train the classification network on the remaining 40,000 noisy samples and evaluate the model on 1,0000 testing images.

CIFAR-100 (Krizhevsky et al, 2009) is more challenging than CIFAR-10 including 100 classes belonging to 20 superclasses where each category contains 600 images with the resolution of 32 × 32. Similar splitting manners as CIFAR-10 are employed.

Clothing1M (Xiao et al, 2015) is a large-scale dataset that is collected from real-world online shopping websites. It contains 1 million images of 14 categories whose labels are generated based on tags extracting from the surrounding texts and keywords, causing huge label noise. The estimated percentage of corrupted labels is around 38.46%. A portion of clean data is also included in Clothing1M, which has been divided into the training set (903k images), validation set (14k images), and test set (10k images). We select the validation set as the meta dataset and evaluate the performance on the test set. We resize all images to 256 × 256 as in (Shu et al, 2019).

Table 1 Architectures of the classification network, the meta-network $V_\phi(\cdot)$, and the prior-network $H_\omega(\cdot)$

Noise Type	Uniform	Flip	Instance	Real-world
CIFAR-10	WRN-28-10	ResNet-34	ResNet-18	-
CIFAR-100	WRN-28-10	ResNet-34	ResNet-18	-
Clothing-1M	-	-	-	ResNet-50
Food-101N	-	-	-	ResNet-50
ANIMAL-10N	-	-	-	VGG-19

	Output size	Layers
$V_\phi(\cdot)$	512	Input ConCat (sample features, embedding labels)
	512	fully connected, tanh
	Num. of classes	fully connected, Sigmoid to $\boldsymbol{\mu_v}$, $\log \boldsymbol{\sigma_v^2}$
$H_\omega(\cdot)$	512	Input sample features
	512	fully connected, tanh
	Num. of classes	fully connected, Sigmoid to $\boldsymbol{\mu_v}$, $\log \boldsymbol{\sigma_v^2}$

Table 2 Hyperparameters of the classification network in our experiments on different datasets.

Dataset	CIFAR-10	CIFAR-100	Clothing1M	Food-101N	ANIMAL-10N
Sampling Number	2	2	1	1	1
Batch Size	100	100	128	128	128
Optimizer	SGD	SGD	SGD	Adam	Adam
Initial Learning Rate	0.02	0.02	0.02	3e-4	3e-4
Decay Rate	5e-4	5e-4	5e-4	-	-
Total Epoch Number	160	160	10	30	30
Momentum	0.9	0.9	0.9	-	-

Food-101N (Lee et al, 2018) is constructed based on the taxonomy of 101 categories in Food-101 (Bossard et al, 2014). It consists of 310k images collected from Google, Bing, Yelp, and TripAdvisor. The noise ratio for labels is around 20%. We select the validation set of 3824 as the meta-data. Following the testing protocol in (Lee et al, 2018; Zhang et al, 2021b), we learn the model on the training set of 55k images and evaluate it on the testing set of the original Food-101.

ANIMAL-10N (Song et al, 2019) contains human-labeled online images for 5 pairs of animals with confusing appearance. The estimated label noise rate is 8%. There are 50,000 training and 5,000 testing images with the resolution of 64 $\times$ 64. We evaluate our model on the dataset without a clean meta set.

Noise settings. We conduct experiments to study four types of corrupted labels. 1) For *flip noise*, we randomly select a transition class for each class and form the label noise by flipping the label to the transition class with a certain probability ρ. 2) For *uniform noise*, we independently change the label to a random class with a probability of ρ. 3) For *instance-dependent (ID) noise*, we adopt the strategy in (Xia et al, 2020b) to construct the dataset with noise caused by the uncertain annotation of the ambiguous observation. 4) For *real-world noise*, differed from the above synthetic noise, it is introduced at the stage of data collection in real world with diverse forms of noise.

For flip, uniform, and ID noise, we conduct experiments under variant settings of noise ratios on CIFAR-10 & 100, where $\rho \in \{0.2, 0.4, 0.6\}$.

Network architectures. The architecture of the classification network affects the performance. We present the result with different backbones in the following comparison experiments and list the architecture with best performance in Tab. 1. Following the setting in (Ren et al, 2018; Shu et al, 2019; Zhang et al, 2021b), we adopt ResNet-18&32&34 (He et al, 2016), Wide ResNet-28-10 (Zagoruyko and Komodakis, 2016), and ResNet-50 (He et al, 2016) in the following experiments. Note that ResNet-32 is a tiny model which is much slimmer than ResNet-18/34. We implement the meta-network and prior network as the three-layer fully-connected network whose dimension for hidden layers is set as 1024. Since its inputs are the feature embedding concatenated with the one-hot label vector, the input dimension is $k+c$, where k, c are the dimension of the feature embedding and the number of categories, respectively. Besides, the dimension of the output layer of the meta-network is $2c$. For the meta-network of $V_\phi(\cdot)$ and the prior-network of $H_\omega(\cdot)$, all models share the same architecture, as in Tab. 1.

Other hyperparameters. The weight coefficient λ for the KL term is set to be 0.001 for all experiments. Its sensitivity for the generalization performance is analysed in the ablation study. The sampling number of k is set as 2 for CIFAR-10 & 100 and 1 for Clothing1M, Food-101N, and

Table 3 Testing Accuracy (%) on CIFAR-10 and CIFAR-100 with varying ratios of three noise types, including **flip noise**, **instance-dependent noise** and **uniform noise**. Note that "ResNet-18/34" denotes applying ResNet-18 for CIFAR-10 and ResNet-34 for CIFAR-100.

Dataset			CIFAR-10		CIFAR-100	
Flip noise						
Noise Ratio			20%	40%	20%	40%
Baseline		ResNet-32	76.83 ± 0.3	70.77 ± 2.3	50.86 ± 0.3	43.01 ± 1.2
MW-Net (Shu et al, 2019)	(NeurIPS19)	ResNet-32	90.33 ± 0.6	87.54 ± 0.2	64.22 ± 0.3	58.64 ± 0.5
MLC (Wang et al, 2020)	(CVPR20)	ResNet-32	90.07 ± 0.2	88.97 ± 0.5	64.91 ± 0.4	59.96 ± 0.6
CORES* (Cheng et al, 2021)	(ICLR21)	ResNet-32	91.41 ± 0.4	89.47 ± 0.3	64.82 ± 0.5	62.76 ± 0.4
PMW-Net (Zhao et al, 2023)	(TNNLS23)	ResNet-32	90.47 ± 0.1	87.69 ± 0.3	64.95 ± 0.2	58.72 ± 0.2
WarPI (Sun et al, 2022)	(PR22)	ResNet-32	90.93	89.87	65.52	62.37
FaMUS (Xu et al, 2021b)	(CVPR21)	ResNet-32	90.78	88.91	65.79	59.66
VRI (Ours)		ResNet-32	**91.93 ± 0.1**	**91.21 ± 0.3**	**66.03 ± 0.2**	**65.04 ± 0.4**
DivideMix (Li et al, 2020)	(ICLR20)	ResNet-18	-	93.4	-	72.1
ELR (Liu et al, 2020)	(NeurIPS20)	ResNet-34	93.28 ± 0.2	90.35 ± 0.4	74.20 ± 0.3	**73.73 ± 0.3**
JNPL (Kim et al, 2021)	(CVPR21)	ResNet-34	93.45	90.72	69.95	59.51
SR (Zhou et al, 2021)	(ICCV21)	ResNet-34	89.55 ± 0.3	85.45 ± 0.2	64.79 ± 0.1	49.51 ± 0.6
MSLC (Wu et al, 2021)	(AAAI21)	ResNet-34	94.11	92.48	70.20	69.24
SFT (Wei et al, 2022)	(ECCV22)	ResNet-34	91.53 ± 0.3	89.93 ± 0.5	71.23 ± 0.3	69.29 ± 0.4
GSS-SSL (Yu et al, 2023)	(CVPR23)	ResNet-34	93.42 ± 0.1	91.82 ± 0.1	73.81 ± 0.2	65.84 ± 0.2
VRI* (Ours)		ResNet-18	**94.87 ± 0.2**	**93.97 ± 0.3**	**76.41 ± 0.3**	68.86 ± 0.3
Instance-dependent noise						
Noise Ratio			20%	40%	20%	40%
Baseline		ResNet-18 / 34	85.10 ± 0.6	77.00 ± 2.1	52.19 ± 1.4	42.26 ± 1.2
Co-teaching (Han et al, 2018b)	(NeurIPS18)	ResNet-18 / 34	86.54 ± 0.1	79.98 ± 0.3	57.24 ± 0.6	45.69 ± 0.9
Peer loss (Liu and Guo, 2020)	(ICML20)	ResNet-18 / 34	88.19 ± 0.5	81.53 ± 0.7	63.82 ± 0.3	47.91 ± 0.5
CORES* (Cheng et al, 2021)	(ICLR21)	ResNet-18 / 34	89.67 ± 0.3	82.99 ± 0.5	64.86 ± 0.5	49.62 ± 0.7
WarPI (Sun et al, 2022)	(PR22)	ResNet-18 / 34	89.76 ± 0.4	87.57 ± 0.9	65.08 ± 0.6	57.38 ± 1.0
CDR (Xia et al, 2020a)	(ICLR21)	ResNet-18 / 34	90.41 ± 0.3	83.07 ± 1.3	67.33 ± 0.6	55.94 ± 0.5
Me-Momen. (Bai and Liu, 2021)	(ICCV21)	ResNet-18 / 34	90.86 ± 0.2	86.66 ± 0.9	68.11 ± 0.5	58.58 ± 1.2
FaMUS (Xu et al, 2021b)	(CVPR21)	ResNet-18 / 34	91.23 ± 0.3	89.88 ± 0.6	66.65 ± 0.5	57.21 ± 1.2
PES (Bai et al, 2021)	(NeurIPS21)	ResNet-18 / 34	92.69 ± 0.4	89.73 ± 0.5	70.49 ± 0.7	65.68 ± 1.4
SFT (Wei et al, 2022)	(ECCV22)	ResNet-18 / 34	91.41 ± 0.3	89.97 ± 0.5	71.83 ± 0.4	69.91 ± 0.5
Late Stop (Yuan et al, 2023)	(ICCV23)	ResNet-18 / 34	91.08 ± 0.2	87.41 ± 0.4	68.59 ± 0.7	59.28 ± 0.5
PADDLES (Huang et al, 2023)	(ICCV23)	ResNet-18 / 34	92.76 ± 0.3	89.87 ± 0.5	70.88 ± 0.6	66.11 ± 1.2
VRI (Ours)		ResNet-18	92.13 ± 0.3	90.60 ± 0.4	71.24 ± 0.2	68.17 ± 0.5
VRI* (Ours)		ResNet-18	**93.36** ± 0.3	**92.96 ± 0.5**	**75.74 ± 0.2**	**70.39 ± 0.6**
Uniform noise						
Noise Ratio			40%	60%	40%	60%
ELR (Liu et al, 2020)	(NeurIPS20)	ResNet-34	91.43 ± 0.2	88.87 ± 0.2	68.43 ± 0.4	60.05 ± 0.9
MSLC (Wu et al, 2021)	(AAAI21)	ResNet-34	91.42	87.25	68.70	60.25
FaMUS (Xu et al, 2021b)	(CVPR21)	ResNet-18	90.50	85.80	69.40	62.90
SFT (Wei et al, 2022)	(ECCV22)	ResNet-18	89.54 ± 0.3	-	69.72 ± 0.3	-
SOP (Liu et al, 2022)	(ICML22)	ResNet-34	90.09 ± 0.3	86.78 ± 0.2	70.12 ± 0.5	60.06 ± 0.4
VRI (Ours)		ResNet-18	91.29 ± 0.2	87.68 ± 0.3	68.92 ± 0.2	62.12 ± 0.2
VRI* (Ours)		ResNet-18	**92.47 ± 0.2**	**89.23 ± 0.3**	**70.45 ± 0.2**	**63.58 ± 0.2**
Baseline		WResNet-28-10	68.07 ± 1.2	53.12 ± 3.0	51.11 ± 0.4	30.92 ± 0.3
MentorNet (Jiang et al, 2018)	(ICML18)	WResNet-28-10	87.33 ± 0.2	82.80 ± 1.4	61.39 ± 4.0	36.87 ± 1.5
MW-Net (Shu et al, 2019)	(NeurIPS19)	WResNet-28-10	89.27 ± 0.3	84.07 ± 0.3	67.73 ± 0.3	58.75 ± 0.1
MLC (Wang et al, 2020)	(CVPR20)	WResNet-28-10	89.20 ± 0.1	84.22 ± 0.3	-	-
DMI-NS (Chen et al, 2021a)	(AAAI21)	WResNet-28-10	91.11 ± 0.5	83.46 ± 0.5	66.95 ± 0.2	58.35 ± 0.1
WarPI (Sun et al, 2022)	(PR22)	WResNet-28-10	89.73	84.44	67.90	59.04
VRI (Ours)		WResNet-28-10	91.29 ± 0.2	84.68 ± 0.2	67.92 ± 0.2	59.32 ± 0.3
VRI* (Ours)		WResNet-28-10	**93.91 ± 0.2**	**91.10 ± 0.2**	**74.95 ± 0.3**	**68.56 ± 0.4**

Table 4 Testing Accuracy (%) on **real-world noise**, including Clothing1M and Food-101N.

Clothing1M				Food-101N			
MWNet (Shu et al, 2019)	73.72	DivideMix (Li et al, 2020)	74.76	Base Model	81.67	CNet_h (Lee et al, 2018)	83.47
ELR (Liu et al, 2020)	74.81	CAL (Zhu et al, 2021)	74.17	MWNet (Shu et al, 2019)	84.72	SMP (Han et al, 2019)	85.11
PLC (Zhang et al, 2021b)	73.24	WarPI (Sun et al, 2022)	74.98	NRank (Sharma et al, 2020)	85.20	ELR+ (Liu et al, 2020)	85.77
JNPL (Kim et al, 2021)	74.15	CoDis (Xia et al, 2023)	74.92	PLC (Zhang et al, 2021b)	85.28	WarPI (Sun et al, 2022)	85.91
NCR (Iscen et al, 2022)	74.42	**VRI** (Ours)	**75.19**	CoDis (Xia et al, 2023)	86.13	**VRI** (Ours)	**86.24**

ANIMAL-10N. For the prior and meta networks, we select the Adam optimizer and set the learning rate as 3e-4 for all experiments. We adopt the CosineAnnealing strategy for adjusting the learning rate of the classification network on CIFAR-10 & 100. Settings of other hyperparameters for the classification network are listed in Tab. 2.

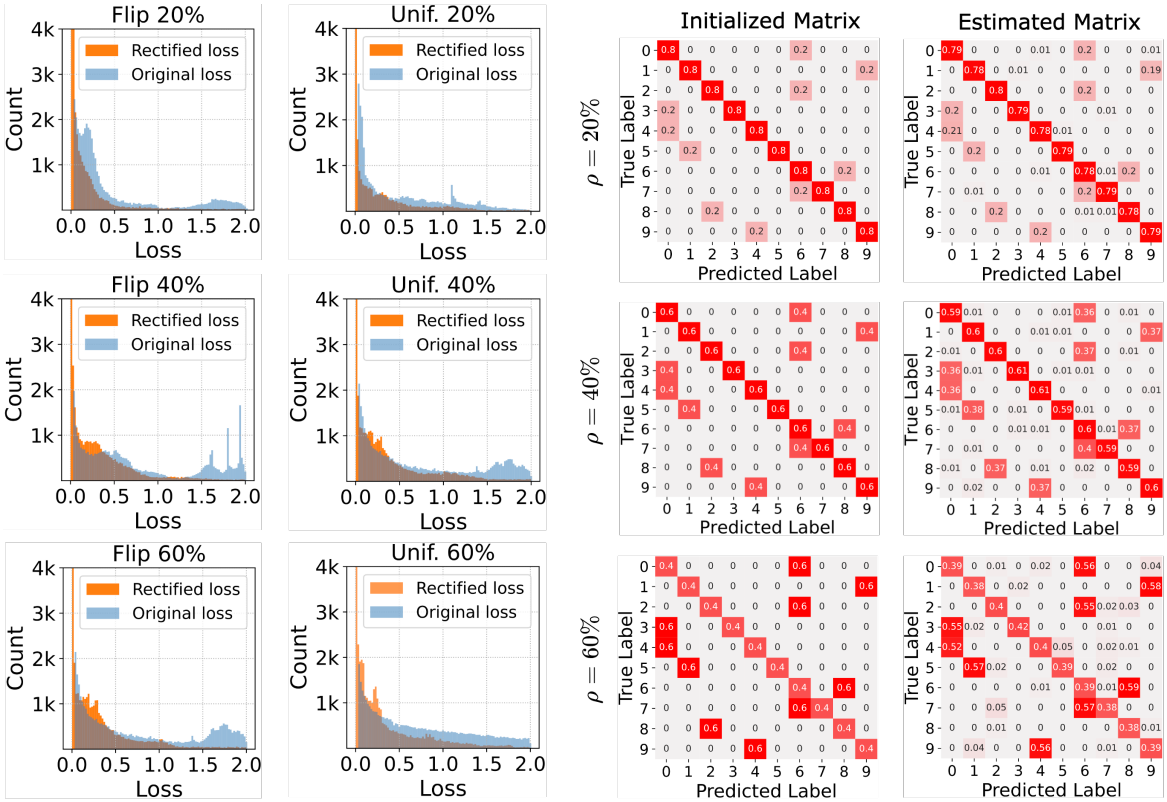
4.2 Comparison results

Synthetic Noise. We evaluate the model on two basic benchmark datasets, *i.e.*, CIFAR-10 and CIFAR-100 of classification tasks. We study variant settings of types and ratios of label noise. For flip & ID noise, we present results with the setting of 20% and 40% noise ratios. For uniform noise, we choose a more challenging setting of 40% and 60% ratios. For a fair comparison, the settings of generating noisy data and network architectures are consistent for all methods. The comparison baseline methods include Base Model that is directly trained on corrupted data, other prevailing approaches (*e. g.*, DivideMix (Li et al, 2020), ELR (Liu et al, 2020), MentorNet (Jiang et al, 2018), CORES* (Cheng et al, 2021), DMI-NS (Chen et al, 2021a), SFT (Wei et al, 2022), GSS-SSL (Yu et al, 2023), PES (Bai et al, 2021), CoDis (Xia et al, 2023), NCR (Iscen et al, 2022), SOP (Liu et al, 2022), Late Stopping (Yuan et al, 2023), PADDLES (Huang et al, 2023) and Me-Momentum (Bai and Liu, 2021)), and meta-learning methods including MSLC (Wu et al, 2021), MW-Net (Shu et al, 2019), PMW-Net (Zhao et al, 2023), MLC (Wang et al, 2020) and FaMUS (Xu et al, 2021b). Note that other works (Li et al, 2019) with fewer fixed transition patterns for flip noise have not been included. To illustrate the effectiveness of variational form of learning to rectifying loss functions, we also compare the method with the homogeneous MC approximation model, WarPI (Sun et al, 2022). To further boost the performance, we adopt the architecture of two models and ensemble them at the testing stage, which is denoted as VRI*.

As shown in Tab. 3, VRI outperforms SOTA meta-learning methods on the classification task and achieves superior performance under the setting of flip noise with ResNet-32. By using boosting techniques, we highlight that VRI* achieves the best performance on three types noise on variant ratios. We would like to highlight that our method gains significant improvement of **4.56**% on CIFAR-100 with 20% ID noise compared with the SOTA method of PADDLES. Besides, VRI consistently outperforms the homologous method of WarPI, indicating the superiority of our variational modeling.

Real-world Noise. To evaluate the performance on real-world noise, we conduct experiments on two large-scale real-world datasets, *i.e.*, Clothing1M and Food-101N, and choose the clean validation set as meta-data. For the fair comparison, we adopt the same evaluation protocol in (Shu et al, 2019; Zhang et al, 2021b) and use the same backbone of ResNet-50 pre-trained on ImageNet. We compare VRI with current SOTA methods. As shown in Tab. 4, the proposed VRI achieves the highest accuracy of 75.19% on Clothing1M and 86.24% on Food-101N, consistently outperforming the homogeneous MC approximation method (*e.g.*, WarPI). VRI also gain a large improvement of 1.4% on Clothing1M and 1.5% on Food-101N compared with other meta-learning methods (*e.g.*, MW-Net), demonstrating its great effectiveness in real-world application.

Indeed, even for the state-of-the-art methods (*e.g.*, DivideMix, ELR), they inevitably involve hyper-parameters and require a clean set (10% of training data, 5k samples of CIFAR) for cross-validation (CV). Our method is proposed to learn an adaptive rectifying strategy in a data-driving way, resolving the issue of *scalability* in CV.



(a) The loss distribution of training data

(b) Estimation of noise transition matrix

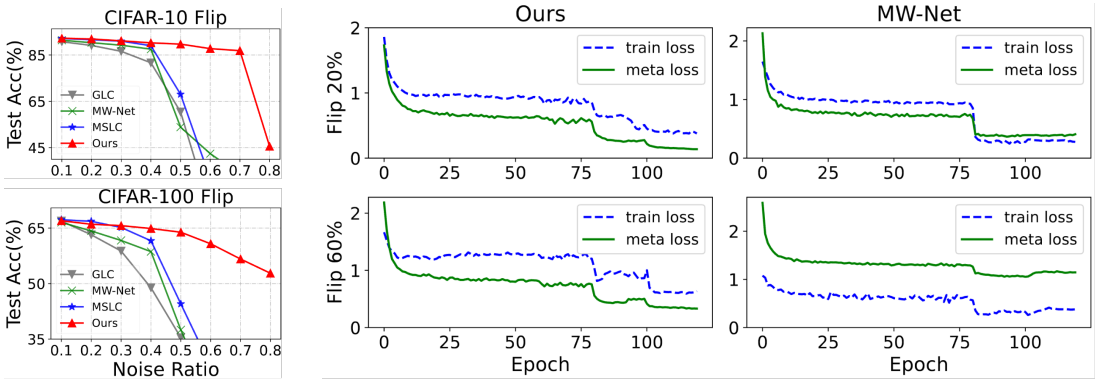
Fig. 4 (a) As the noise ratio increases, the effect of rectification becomes more obvious since the area of the original loss increases. (b) We almost achieve the unbiased estimation for the initialized transition matrix of flip noise with varying noise ratio ρ.

4.3 Further Analysis

Effectiveness. To directly visualize the effect after rectification, we plot the distribution of training losses for all samples in Fig. 4 (a) when finishing the training process. The blue part represents the original loss without rectification, while the orange is for the loss computed from the rectified logits using our meta-network. As shown in Fig. 4 (a), the rectified loss is lower than the original one with high probability. The area of the original loss increases as the noise ratio rises, indicating the effect of rectification becomes more obvious. To further illustrate its effectiveness, we adopt the prediction from the rectified logits as the clean label to estimate the transition matrix for constructing flip noise. We draw the initialized and estimated transition matrices for 20%, 40%, and 60% ratios on CIFAR-10 in Fig. 4 (b). We almost achieve the unbiased estimation for the initialized matrix.

Robustness. We evaluate the generalization ability of VRI on more challenging conditions with high flip noise ratios. We compare VRI with three typical meta-learning methods, *i.e.*, GLC (Hendrycks et al, 2018) of loss correction, MW-Net (Shu et al, 2019) of reweighting, MSLC (Wu et al, 2021) of label correction. We adopt the same backbone network of ResNet-32 and a consistent setting of 1,000 meta samples. As shown in Fig. 5 (a), VRI can still produce favorable results, even on the challenging condition with a far higher noise ratio. Compared with the SOTA meta-learning methods, VRI can retain the high accuracy of 86% on CIFAR-10 with the setting of 70% noise ratio.

We also plot Fig. 5 (b) about the training and meta loss to explain this phenomenon. For other meta-learners (*e. g.*, MW-Net), their meta-network might have limited ability to conduct the meta-learning process with a high ratio of flip noise. As the noise rate exceeds 50%, the



(a) Extreme noise ratio ρ (b) Convergence comparison with MW-Net

Fig. 5 (a) The performance of VRI and other three typical meta-learning methods as the noise ratio increases. Our method can still produce good performance with a far higher noise ratio. (b) Our algorithm achieves a stable convergence and displays robustness on flip noise with a high ratio (e.g., 60%).

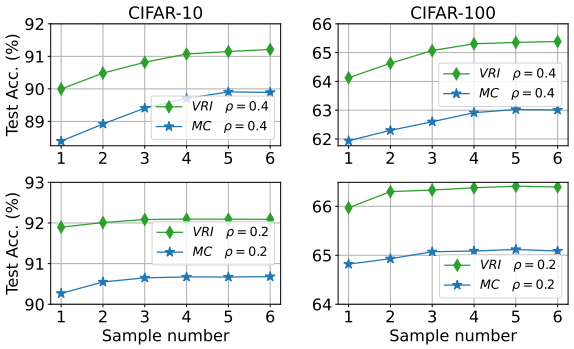


Fig. 6 MC method is more sensitive about the sample number compared with our proposed VRI.

learning process in MW-Net is dominated by the classification network, where the empirical error decreases rapidly but the meta error still keeps high. This renders non-convergence for optimizing the meta-network, leading to poor generalization performance. For MSLC, the backbone needs warm-up with the training data, which certainly degenerates the performance for high noise ratios. For VRI, our meta-network is powerful enough to rectify the training process by taking the feature and label as input and generating an effective rectifying vector, which is endowed with robustness to flip noise with high ratios.

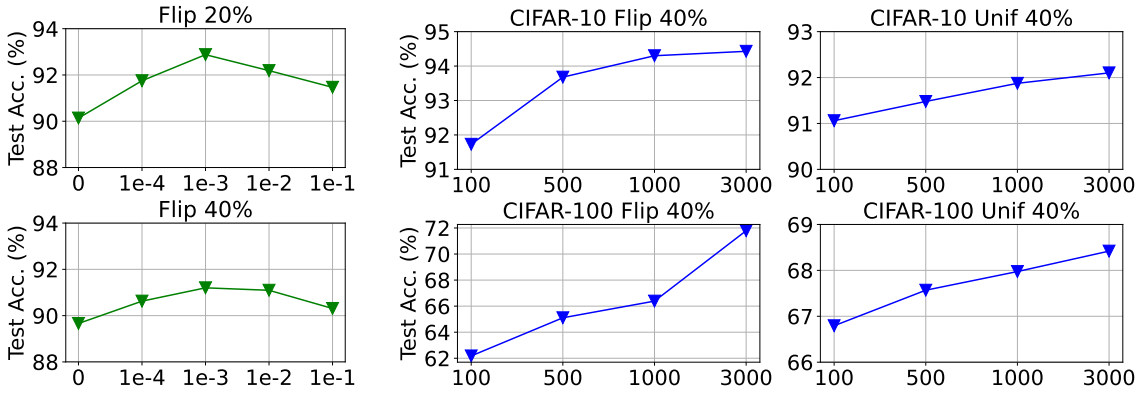
4.4 Ablation Study

Sampling number. The sampling number k in Monte Carlo (MC) approximation has an impact on performance. We conduct experiments on CIFAR-10 and CIFAR-100 with variant k for two flip noise ratios. As shown in Fig. 6, the testing accuracy for MC essentially turns to be higher, then keeps stable as the sample number increases. Despite the gain of the performance from more samples, the training time increases linearly as illustrated in Tab. 5. Thanks to the variational term in VRI, we achieve higher accuracy than the MC approximation while keeping good efficiency.

Table 5 VRI yields higher performance than MC approximation with an efficient inference.

	k	Time (min./epoch)	Test Acc. (%)
MC	1	2.17	88.23
	3	4.32	89.45
	5	7.04	89.87
VRI	1	2.20	90.20

Hyper-parameter discussion. To illustrate the sensitivity of λ, we conduct experiments on CIFAR-10 under flip noise. As shown in Fig. 7 (a), we obtain the best performance with $\lambda = 0.001$. The accuracy would slightly drop as λ increases. Indeed, we observe that the KL divergence of the variational term usually produces a large value at the beginning of the training, which would lead to an unstable learning process. Therefore, we set λ as 0.001 via cross-validation. The result also demonstrates that we can gain considerable improvement of the performance by introducing the variational term.



(a) Sensitivity analysis of λ (b) Sensitivity analysis of the number of the meta set

Fig. 7 (a) We obtain the best performance on with $\lambda = 0.001$. (b) The performance improves as the number of meta samples increases.

Table 6 Testing Accuracy (%) on CIFAR-10 and CIFAR-100 with **uniform noise** (top) and **flip noise** (bottom) when the accessibility of meta-data is restricted.

	Dataset		CIFAR-10		CIFAR-100		
	Noise Ratio	Structure	40%	60%	40%	60%	Average gap
Uniform	Baseline	ResNet-18	68.07±1.23	53.12±3.03	51.11±0.42	30.92±0.33	26.0
	VRI (-)	ResNet-18	80.23±1.42	74.54±2.46	59.39±0.73	49.39±0.46	10.1
	VRI (+)	ResNet-18	91.24±1.42	87.45±2.16	66.39±0.44	58.60±0.56	1.0
	VRI (Aug.)	ResNet-18	90.78±1.56	87.98±2.12	66.78±0.68	58.79±0.76	0.8
	VRI (Standard)	ResNet-18	**91.58±0.17**	**88.68±0.22**	**67.92±0.19**	**59.32±0.31**	0
Flip	Baseline	ResNet-32	76.83±0.32	70.77±2.31	50.86±0.27	43.01±1.16	18.2
	VRI (-)	ResNet-32	82.23±1.06	80.34±1.96	58.47±0.78	55.78±0.45	9.4
	VRI (+)	ResNet-32	90.88±1.16	90.36±1.84	65.47±0.81	64.36±0.55	0.8
	VRI (Aug.)	ResNet-32	91.11±1.12	90.34±1.87	65.67±0.98	64.24±0.56	0.5
	VRI (Standard)	ResNet-32	**91.93±0.14**	**91.21±0.33**	**66.03±0.21**	**65.04±0.38**	0

The cardinality of the meta set. The cardinality of the meta set has an impact on the performance. We set it to 1,000 for CIFAR datasets as other meta-learning methods (*e.g.*, MWNet, MSLC). We also study the influence in Fig. 7 (b). The performance improves as the number of meta samples increases, especially for flip noise. Also, VRI can obtain considerable performance (91.07%, CIFAR-10, flip 40%) given limited meta samples (100). Here, the backbone is ResNet-18.

4.5 Learning without Meta-Data

To evaluate the performance of the model when there is a lack of clean meta-data, we adopt the sample selection strategy (Han et al, 2018b) to select reliable samples in the corrupted training set and treat them as pseudo meta-data. Specifically, we firstly conduct warming-up (CIFAR-10: 10 epochs. CIFAR-100: 30 epochs. ANIMAL-10N: 100 epochs) for the classification network to achieve the basic discrimination ability. Then, we apply the small-loss strategy and select 1,000 samples with a higher confidence for each epochs. Next, we train our meta-network with the selected samples by using the proposed learning Algorithm 1. The whole process can be summarized as Algorithm 2 in the Appendix A.3.

For synthetic noise, the class distribution has an impact on the performance. We conduct two experiments. a) "**VRI** (+)", balancing the class of selected meta data; b) "**VRI** (−)", directly using the selected samples with the top 1,000 smallest losses. We observe that classes of the latter are

Table 7 Testing Accuracy (%) of VRI without given meta-data on ANIMAL-10N.

Baseline	Song et al (2019)	Zhang et al (2021b)	Chen et al (2021b)	Englesson (2021)	Chen et al (2022)	**VRI** (-)	**VRI** (+)
79.4	81.8	83.4	84.1	84.2	84.5	81.4	**85.8**

extremely imbalanced. Besides, data augmentation can also relieve the class-imbalance issue. We select all training samples with the smaller loss via Gaussian Mixture Model clustering and use mixup to enhance training / meta-data.

As shown in Tab. 6, the performance heavily degenerates with imbalanced pseudo meta-data. Besides, the meta-learning framework without meta-data still outperforms the baseline that is directly trained on the noisy dataset and achieves favorable performance.

For real-world noise, VRI achieves the highest accuracy without meta data on the ANIMAL-10N dataset (Tab. 7). We adopt the same architecture of VGG19 as (Song et al, 2019; Zhang et al, 2021b; Chen et al, 2021b). To build the meta set, We firstly train the VGG19 for 100 epochs in a standard manner. We then use this network to select clean samples with the top 1,000 smallest empirical losses as meta data and carefully balance the number (100) for each class. Once we split the original training set into the noisy training set and meta set, we meta-learn a new VGG19 network from *scratch* via VRI for evaluation.

5 Conclusion

In this work, we propose variational rectification inference (VRI) for learning with label noise to tackle model collapse in the MC meta-learning method. VRI is built as a hierarchical Bayes to estimate the conditional predictive distribution and formulated as the variational inference problem. To achieve adaptively rectifying the loss with noisy labels, we design a meta-network, which is endowed with the ability to exploit information lying in the feature space. Our method can also meta-learn the rectifying process via bi-level programming, whose convergence can be theoretically guaranteed. To evaluate the effectiveness of VRI, we conduct extensive experiments on varied noise types and achieve competitive performance on those benchmarks. Experimental results demonstrate that VRI outperforms the MC method with low sampling rates, resulting in a more efficient learning process and resolving the issue of *scalability* in cross-validation. To further boost our framework, we integrate the adaptive sample strategy into VRI and obtain comparable performance without meta data, beyond the common setting of existing meta-learning methods.

Appendix A

A.1 Derivations of The ELBO

For a singe observation $(\mathbf{x}, \mathbf{y})$, the ELBO can be derived from the perspective of the KL divergence between the variational posterior $q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y})$ and the posterior $p(\mathbf{v}|\mathbf{x}, \mathbf{y})$:

$$
\begin{aligned}
&D_{\mathrm{KL}}[q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y})||p(\mathbf{v}|\mathbf{x}, \mathbf{y})] \\
&= \mathbb{E}_{q_\phi(\mathbf{v}|\mathbf{x},\mathbf{y})}\left[\log q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y}) - \log p(\mathbf{v}|\mathbf{x}, \mathbf{y})\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{v}|\mathbf{x},\mathbf{y})}\left[\log q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y}) - \log \frac{p(\mathbf{v}|\mathbf{x}, \mathbf{y})p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x}, \mathbf{y})}\right] \\
&= \log p(\mathbf{y}|\mathbf{x}) + \mathbb{E}_{q_\phi(\mathbf{v}|\mathbf{x},\mathbf{y})}\big[\log q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y}) \\
&\qquad\qquad - \log p(\mathbf{y}|\mathbf{x}, \mathbf{v}) - \log p(\mathbf{v}|\mathbf{x})\big] \\
&= \log p(\mathbf{y}|\mathbf{x}) - \mathbb{E}_{q_\phi(\mathbf{v}|\mathbf{x},\mathbf{y})}\left[\log p(\mathbf{y}|\mathbf{x}, \mathbf{v})\right] \\
&\qquad\qquad + D_{\mathrm{KL}}[q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y})||p(\mathbf{v}|\mathbf{x})] \\
&\geq 0.
\end{aligned}
\tag{A1}
$$

Specifically, we apply Bayes' rule to derive Eq. (A1) as

$$
\begin{aligned}
p(\mathbf{v}|\mathbf{x}, \mathbf{y}) &= \frac{p(\mathbf{v}|\mathbf{x}, \mathbf{y})p(\mathbf{x}, \mathbf{y})}{p(\mathbf{x}, \mathbf{y})} \\
&= \frac{p(\mathbf{y}|\mathbf{x}, \mathbf{v})p(\mathbf{x}, \mathbf{v})}{p(\mathbf{x}, \mathbf{y})} = \frac{p(\mathbf{y}|\mathbf{x}, \mathbf{v})p(\mathbf{v}|\mathbf{x})}{p(\mathbf{y}|\mathbf{x})}.
\end{aligned}
\tag{A2}
$$

Therefore, the ELBO for the log-likelihood of the predictive distribution in Eq. (3) can be written as follows

$$
\begin{aligned}
&\log p(\mathbf{y}|\mathbf{x}) \\
&\geq \mathbb{E}_{q_\phi(\mathbf{v}|\mathbf{x},\mathbf{y})}\left[\log p(\mathbf{y}|\mathbf{x}, \mathbf{v})\right] - D_{\mathrm{KL}}[q_\phi(\mathbf{v}|\mathbf{x}, \mathbf{y})||p(\mathbf{v}|\mathbf{x}))] \\
&= \mathcal{L}_{\mathrm{ELBO}}.
\end{aligned}
\tag{A3}
$$

A.2 Proof

Lemma 1 (Smoothness)

Proof. We begin with computation of the derivation of the meta loss $\widetilde{\mathcal{L}}^{emp}(\hat{\theta})$ *w.r.t.* the meta network ϕ. By using Eq. (9), we have

$$\begin{aligned}\frac{\partial \mathcal{L}^{meta}(\hat{\theta})}{\partial \phi} &= \frac{\partial \mathcal{L}^{meta}(\hat{\theta})}{\partial \hat{\theta}} \frac{\partial \hat{\theta}}{\partial V(\phi)} \frac{\partial V(\phi)}{\partial \phi} \\ &= \alpha \frac{\partial \mathcal{L}^{meta}(\hat{\theta})}{\partial \hat{\theta}} \left(\nabla_\theta L(\theta) + \frac{\partial D_{\mathrm{KL}}}{\partial V(\phi)} \right) \frac{\partial V(\phi)}{\partial \phi}. \end{aligned} \tag{A4}$$

To simplify the proof, we neglect Monte Carlo estimation in Eq. 6 and consider it as a deterministic rectified vector in the following. This would not affect the result since there ultimately exists a rectified vector for computing the expectation of those sampled losses. Taking the gradient of ϕ on both side of Eq. (A4),

$$\begin{aligned}&\frac{\partial^2 \mathcal{L}^{meta}(\hat{\theta})}{\partial \phi^2} \\ &= \underbrace{\alpha \frac{\partial}{\partial \phi} \left(\frac{\partial \mathcal{L}^{meta}(\hat{\theta})}{\partial \hat{\theta}} \left(\nabla_\theta L(\theta) + \frac{\partial D_{\mathrm{KL}}}{\partial V(\phi)} \right) \right) \frac{\partial V(\phi)}{\partial \phi}}_{❶} \\ &+ \underbrace{\alpha \frac{\partial \mathcal{L}^{meta}(\hat{\theta})}{\partial \hat{\theta}} \left(\nabla_\theta L(\theta) + \frac{\partial D_{\mathrm{KL}}}{\partial V(\phi)} \right) \frac{\partial^2 V(\phi)}{\partial \phi^2}}_{❷}. \end{aligned}$$

For the first term ❶ in the right hand, we can obtain the following inequality *w.r.t.* its norm

$$\begin{aligned}\|❶\| &\le \alpha\delta \left\| \frac{\partial}{\partial \hat{\theta}} \left(\frac{\partial \mathcal{L}^{meta}(\hat{\theta})}{\partial \phi} \right) \left(\nabla_\theta L(\theta) + \frac{\partial D_{\mathrm{KL}}}{\partial V(\phi)} \right) \right\| \\ &= \alpha^2 \delta \left\| \frac{\partial}{\partial \hat{\theta}} \left(\frac{\partial \mathcal{L}^{meta}(\hat{\theta})}{\partial \hat{\theta}} \left(\nabla_\theta L(\theta) + \frac{\partial D_{\mathrm{KL}}}{\partial V(\phi)} \right) \frac{\partial V(\phi)}{\partial \phi} \right) \right. \\ &\qquad \left. \left(\nabla_\theta L(\theta) + \frac{\partial D_{\mathrm{KL}}}{\partial V(\phi)} \right) \right\| \\ &= \alpha^2 \delta \left\| \frac{\partial^2 \mathcal{L}^{meta}(\hat{\theta})}{\partial \hat{\theta}^2} \left(\nabla_\theta L(\theta) + \frac{\partial D_{\mathrm{KL}}}{\partial V(\phi)} \right)^2 \frac{\partial V(\phi)}{\partial \phi} \right\| \\ &\le \ell \alpha^2 \delta^2 (\tau + o)^2, \end{aligned}$$

since we assume $\|\frac{\partial^2 \mathcal{L}^{meta}(\hat{\theta})}{\partial \hat{\theta}^2}\| \le \ell$, $\|\nabla_\theta L(\theta)\| \le \tau$, $\|\frac{\partial D_{\mathrm{KL}}}{\partial V(\phi)}\| \le o$, and $\|\frac{\partial V(\phi)}{\partial \phi}\| \le \delta$.

For the second term ❷, we can also obtain

$$\|❷\| \le \alpha\tau(\tau + o)\zeta$$

with the assumption $\|\frac{\partial^2 V(\phi)}{\partial \phi^2}\| \le \zeta$. Therefore, we have

$$\left\| \frac{\partial^2 \mathcal{L}^{meta}(\hat{\theta})}{\partial \phi^2} \right\| \le \alpha(\tau + o) \left(\ell\alpha\delta^2(\tau + o) + \tau\zeta \right).$$

Let $\hat{\ell} = \alpha(\tau + o) \left(\ell\alpha\delta^2(\tau + o) + \tau\zeta \right)$, we can conclude the proof that

$$\|\mathcal{L}^{meta}(\hat{\theta}(\phi^{(t+1)})) - \mathcal{L}^{meta}(\hat{\theta}(\phi^{(t)}))\| \le \hat{\ell} \|\phi^{(t+1)} - \phi^{(t)}\|.$$

□

Theorem 1 (Convergence Rate)

Proof. Consider

$$\begin{aligned}&\mathcal{L}^{meta}(\hat{\theta}^{(t+1)}(\phi^{(t+1)})) - \mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)})) \\ &= \underbrace{\mathcal{L}^{meta}(\hat{\theta}^{(t+1)}(\phi^{(t+1)})) - \mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t+1)}))}_{❸} \\ &\quad + \underbrace{\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t+1)})) - \mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)}))}_{❹}. \end{aligned}$$

For ❸, by Lipschitz smoothness of the meta loss function for θ, we have

$$\begin{aligned}&\mathcal{L}^{meta}(\hat{\theta}^{(t+1)}(\phi^{(t+1)})) - \mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t+1)})) \\ &\le \langle \nabla \mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t+1)})), \hat{\theta}^{(t+1)}(\phi^{(t+1)}) - \hat{\theta}^{(t)}(\phi^{(t+1)}) \rangle \\ &\quad + \frac{\ell}{2} \|\hat{\theta}^{(t+1)}(\phi^{(t+1)}) - \hat{\theta}^{(t)}(\phi^{(t+1)})\|_2^2. \end{aligned} \tag{A5}$$

We firstly write $\hat{\theta}^{(t+1)}(\phi^{(t+1)}), \hat{\theta}^{(t)}(\phi^{(t+1)})$ with Eq. (9). Using Eq. (12), we obtain

$$\begin{aligned}&\hat{\theta}^{(t+1)}(\phi^{(t+1)}) - \hat{\theta}^{(t)}(\phi^{(t+1)}) \\ &\quad = -\alpha \nabla_\theta \mathcal{L}^{emp}(\hat{\theta}^{(t+1)}(\phi^{(t+1)})). \end{aligned} \tag{A6}$$

and

$$\begin{aligned}&\|\mathcal{L}^{meta}(\hat{\theta}^{(t+1)}(\phi^{(t+1)})) - \mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t+1)}))\| \\ &\quad \le \alpha_t \tau^2 + \frac{\ell \alpha_t^2}{2} \tau^2 = \alpha_t \tau^2 (1 + \frac{\alpha_t \ell}{2}), \end{aligned} \tag{A7}$$

since $\left\|\frac{\partial L(\theta)}{\partial \theta}\Big|_{\theta^{(t)}}\right\| \leq \tau$, $\left\|\frac{\partial L_i^{meta}(\hat{\theta})}{\partial \hat{\theta}}\Big|_{\hat{\theta}^{(t)}}\right\| \leq \tau$, and the output of $V(\cdot)$ is bounded with the sigmoid function.

For ❹, since the gradient is computed from a mini-batch of training data that is drawn uniformly, we denote the bias of the stochastic gradient $\varepsilon^{(t)} = \nabla\widetilde{\mathcal{L}}^{meta}\left(\hat{\theta}^{(t)}\left(\phi^{(t)}\right)\right) - \nabla\mathcal{L}^{meta}\left(\hat{\theta}^{(t)}\left(\phi^{(t)}\right)\right)$. We then observe its expectation obeys $\mathbb{E}[\varepsilon^{(t)}] = 0$ and its variance obeys $\mathbb{E}[\|\varepsilon^{(t)}\|_2^2] \leq \sigma^2$.

By smoothness of $\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi))$ for ϕ in Lemma 1, we have

$$\begin{aligned}
&\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t+1)})) - \mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)})) \\
&\leq \langle\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)})), \phi^{(t+1)} - \phi^{(t)}\rangle + \frac{\hat{\ell}}{2}\|\phi^{(t+1)} - \phi^{(t)}\|_2^2 \\
&= \langle\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)})), -\eta_t[\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)})) + \varepsilon^{(t)}]\rangle \\
&\quad + \frac{\hat{\ell}\eta_t^2}{2}\|\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)})) + \varepsilon^{(t)}\|_2^2 \\
&= -(\eta_t - \frac{\hat{\ell}\eta_t^2}{2})\|\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)}))\|_2^2 + \frac{\widetilde{\ell}\eta_t^2}{2}\|\varepsilon^{(t)}\|_2^2 \\
&\quad - (\eta_t - \hat{\ell}\eta_t^2)\langle\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)})), \varepsilon^{(t)}\rangle. \qquad \text{(A8)}
\end{aligned}$$

Thus far Eq.(A5) satisfies

$$\begin{aligned}
&\mathcal{L}^{meta}(\hat{\theta}^{(t+1)}(\phi^{(t+1)})) - \mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)})) \\
&\leq \alpha_t\tau^2(1 + \frac{\alpha_t\ell}{2}) - (\eta_t - \frac{\hat{\ell}\eta_t^2}{2})\|\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)}))\|_2^2 \\
&\quad + \frac{\hat{\ell}\eta_t^2}{2}\|\varepsilon^{(t)}\|_2^2 - (\eta_t - \hat{\ell}\eta_t^2)\langle\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)})), \varepsilon^{(t)}\rangle. \qquad \text{(A9)}
\end{aligned}$$

We take the expectation *w.r.t.* $\varepsilon^{(t)}$ over Eq. (A9) and sum up T inequalities. By the property of the bias $\varepsilon^{(t)}$, we can obtain

$$\begin{aligned}
&\sum_{t=1}^{T}\left(\mathop{\mathbb{E}}_{\varepsilon^{(t)}}\mathcal{L}^{meta}(\hat{\theta}^{(t+1)}(\phi^{(t+1)})) - \mathop{\mathbb{E}}_{\varepsilon^{(t)}}\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)}))\right) \\
&\leq \tau^2\sum_{t=1}^{T}\alpha_t(1 + \frac{\alpha_t\ell}{2}) \\
&\quad - \sum_{t=1}^{T}(\eta_t - \frac{\hat{\ell}\eta_t^2}{2})\mathop{\mathbb{E}}_{\varepsilon^{(t)}}\left[\|\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)}))\|_2^2\right] + \frac{\hat{\ell}\sigma^2}{2}\sum_{t=1}^{T}\eta_t^2.
\end{aligned}$$

Taking the total expectation and reordering the terms, we have

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}(\eta_t - \frac{\hat{\ell}\eta_t^2}{2})\,\mathbb{E}\left[\|\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)}))\|_2^2\right] \\
&\leq \frac{\mathcal{L}^{meta}(\hat{\theta}^{(0)}(\phi^{(0)})) - \mathbb{E}\left[\mathcal{L}^{meta}(\hat{\theta}^{(T+1)}(\phi^{(T+1)}))\right]}{T} \\
&\quad + \frac{\tau^2}{T}\sum_{t=1}^{T}\alpha_t(1 + \frac{\alpha_t\ell}{2}) + \frac{\hat{\ell}\sigma^2}{2T}\sum_{t=1}^{T}\eta_t^2. \qquad \text{(A10)}
\end{aligned}$$

Let $E = \mathcal{L}^{meta}(\hat{\theta}^{(0)}(\phi^{(0)})) - \mathbb{E}\left[\mathcal{L}^{meta}(\hat{\theta}^{(T+1)}(\phi^{(T+1)}))\right]$. With the assumption of $\eta_t = \min\{\frac{1}{\hat{\ell}}, \frac{C}{\sigma\sqrt{T}}\}$ and $\alpha_t = \min\{1, \frac{\kappa}{T}\}$, we have $\eta_t - \frac{\hat{\ell}\eta_t^2}{2} \geq \eta_t - \frac{\eta_t}{2} = \frac{\eta_t}{2}$ and

$$\begin{aligned}
&\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\|\nabla\mathcal{L}^{meta}(\hat{\theta}^{(t)}(\phi^{(t)}))\|_2^2\right] \\
&\leq \frac{2E}{T\eta_1} + \frac{(2+\ell)\tau^2\alpha_1}{\eta_1} + \hat{\ell}\sigma^2\eta_1 \\
&= \frac{2E}{T}\max\{\hat{\ell}, \frac{\sigma\sqrt{T}}{C}\} + (2+\ell)\tau^2\min\{1, \frac{\kappa}{T}\}\max\{\hat{\ell}, \frac{\sigma\sqrt{T}}{C}\} \\
&\quad + \hat{\ell}\sigma^2\min\{\frac{1}{\hat{\ell}}, \frac{C}{\sigma\sqrt{T}}\} \\
&\leq \frac{2\sigma E}{C\sqrt{T}} + \frac{(2+\ell)\tau^2\kappa\sigma}{C\sqrt{T}} + \frac{C\hat{\ell}\sigma^2}{\sigma\sqrt{T}} = \mathcal{O}(\frac{1}{\sqrt{T}}).
\end{aligned}$$

Thus, we conclude our proof. □

A.3 Algorithm for VRI without the meta set

Declarations

- **Funding** This research was supported by Natural Science Foundation of China (No. 62106129, 62176139, 62276155), Natural Science Foundation of Shandong Province (No. ZR2021QF053, ZR2021ZD15)
- **Conflict of interest** The author declares that he has no confict of interest.
- **Ethics approval** Not applicable.
- **Consent to participate** Not applicable.
- **Consent for publication** Not applicable.
- **Availability of data and materials** Not applicable.

Algorithm 2 Learning without meta data

Require: Training set $\mathcal{D}_N$, number of meta samples M, batch size n, m, outer iterations T for each epoch, sampling number k, step size α, η, warming-up epoch K, training epoch C

Ensure: Optimal θ^*

Initialize parameters $\theta^{(0)}$, $\phi^{(0)}$, and $\omega^{(0)}$
Warm up parameters θ for K epochs
for $c \in \{1, \ldots, C\}$ **do**
  $\mathcal{D}_M = \text{SelectWithBalance}(\mathcal{D}_N, M)$
  for $t \in \{1, \ldots, T\}$ **do**
    $\text{SampleBatch}(\mathcal{D}_N, n), \text{SampleBatch}(\mathcal{D}_M, m)$
    Form learning process of $\hat{\theta}^{(t)}(\phi, \omega)$
    Optimize $\phi^{(t)}$ with $\hat{\theta}^{(t)}(\phi)$
    Optimize $\omega^{(t)}$ with $\hat{\theta}^{(t)}(\omega)$
    Optimize $\theta^{(t)}$ using the updated $\phi^{(t+1)}$
  end for
end for

- **Code availability** The code is now available at https://github.com/haolsun/VRI.
- **Authors' contributions** H-Sun conceptualized the learning problem and provided the main idea. He also drafted the article. Q-Wei completed main experiments and provided the analysis of experimental results. L-Feng provided the theoretical guarantee for the learning algorithm. F-Liu and H-Fan contributed to participating in discussions of the algorithm and experimental designs. Y-Hu and Y-Yin provided funding supports, and Y-Hu approved the final version of the article.

References

Arazo E, Ortego D, Albert P, et al (2019) Unsupervised label noise modeling and loss correction. In: ICML

Arpit D, Jastrzkebski S, Ballas N, et al (2017) A closer look at memorization in deep networks. In: ICML

Bai Y, Liu T (2021) Me-momentum: Extracting hard confident examples from noisily labeled data. In: ICCV

Bai Y, Yang E, Han B, et al (2021) Understanding and improving early stopping for learning with noisy labels. In: NeurIPS

Bao F, Wu G, Li C, et al (2021) Stability and generalization of bilevel programming in hyperparameter optimization. In: NeurIPS

Berthelot D, Carlini N, Goodfellow I, et al (2019) Mixmatch: A holistic approach to semi-supervised learning. NeurIPS

Bossard L, Guillaumin M, Van Gool L (2014) Food-101–mining discriminative components with random forests. In: ECCV

Chen P, Ye J, Chen G, et al (2021a) Robustness of accuracy metric and its inspirations in learning with noisy labels. In: AAAI

Chen Y, Shen X, Hu SX, et al (2021b) Boosting co-teaching with compression regularization for label noise. In: CVPR

Chen Y, Hu SX, Shen X, et al (2022) Compressing features for learning with noisy labels. IEEE Trans on Neural Networks and Learning Systems (Early Access)

Cheng H, Zhu Z, Li X, et al (2021) Learning with instance-dependent label noise: A sample sieve approach. In: ICLR

Cui Y, Jia M, Lin TY, et al (2019) Class-balanced loss based on effective number of samples. In: CVPR

Englesson E (2021) Generalized jensen-shannon divergence loss for learning with noisy labels. In: NeurIPS

Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks. In: ICML

Franceschi L, Frasconi P, Salzo S, et al (2018) Bilevel programming for hyperparameter optimization and meta-learning. In: ICML

Ge Y, Ren J, Gallagher A, et al (2023) Improving zero-shot generalization and robustness of multi-modal models. In: CVPR

Ghosh A, Kumar H, Sastry P (2017) Robust loss functions under label noise for deep neural networks. In: AAAI

Goldberger J, Ben-Reuven E (2017) Training deep neural-networks using a noise adaptation layer.

In: ICLR

Gudovskiy D, Rigazio L, Ishizaka S, et al (2021) Autodo: Robust autoaugment for biased data with label noise via scalable probabilistic implicit differentiation. In: CVPR

Han B, Yao J, Niu G, et al (2018a) Masking: A new perspective of noisy supervision. In: NeurIPS

Han B, Yao Q, Yu X, et al (2018b) Co-teaching: Robust training of deep neural networks with extremely noisy labels. In: NeurIPS

Han J, Luo P, Wang X (2019) Deep self-learning from noisy labels. In: ICCV

He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: CVPR

Hendrycks D, Mazeika M, Wilson D, et al (2018) Using trusted data to train deep networks on labels corrupted by severe noise. In: NeurIPS

Higgins I, Matthey L, Pal A, et al (2017) beta-vae: Learning basic visual concepts with a constrained variational framework. In: ICLR

Hospedales T, Antoniou A, Micaelli P, et al (2022) Meta-learning in neural networks: A survey. IEEE Trans on Pattern Analysis and Machine Intelligence 44(9):5149–5169

Huang H, Kang H, Liu S, et al (2023) Paddles: Phase-amplitude spectrum disentangled early stopping for learning with noisy labels. In: ICCV

Iakovleva E, Verbeek J, Alahari K (2020) Meta-learning with shared amortized variational inference. In: ICML

Iscen A, Valmadre J, Arnab A, et al (2022) Learning with neighbor consistency for noisy labels. In: CVPR

Jiang L, Zhou Z, Leung T, et al (2018) Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: ICML

Kim Y, Yun J, Shon H, et al (2021) Joint negative and positive learning for noisy labels. In: CVPR

Kingma DP, Welling M (2014) Auto-encoding variational bayes. In: ICLR

Krizhevsky A, Hinton G, et al (2009) Learning multiple layers of features from tiny images. In: Toronto, ON, Canada

Kumar MP, Packer B, Koller D (2010) Self-paced learning for latent variable models. In: NeurIPS

Lee KH, He X, Zhang L, et al (2018) Cleannet: Transfer learning for scalable image classifier training with label noise. In: CVPR

Li J, Wong Y, Zhao Q, et al (2019) Learning to learn from noisy labeled data. In: CVPR

Li J, Socher R, Hoi SC (2020) Dividemix: Learning with noisy labels as semi-supervised learning. In: ICLR

Li S, Xia X, Ge S, et al (2022) Selective-supervised contrastive learning with noisy labels. In: CVPR

Liu H, Zhong Z, Sebe N, et al (2023) Mitigating robust overfitting via self-residual-calibration regularization. Artificial Intelligence 317:103877

Liu S, Niles-Weed J, Razavian N, et al (2020) Early-learning regularization prevents memorization of noisy labels. In: NeurIPS

Liu S, Zhu Z, Qu Q, et al (2022) Robust training under label noise by over-parameterization. In: ICML

Liu Y, Guo H (2020) Peer loss functions: Learning from noisy labels without knowing noise rates. In: ICML

Ma X, Wang Y, Houle ME, et al (2018) Dimensionality-driven learning with noisy labels. In: ICML

Nishi K, Ding Y, Rich A, et al (2021) Augmentation strategies for learning with noisy labels. In: CVPR

Ortego D, Arazo E, Albert P, et al (2021) Multi-objective interpolation training for robustness to label noise. In: CVPR

Pereyra G, Tucker G, Chorowski J, et al (2017) Regularizing neural networks by penalizing

confident output distributions. arXiv preprint arXiv:170106548

Pu N, Zhong Z, Sebe N, et al (2023) A memorizing and generalizing framework for lifelong person re-identification. IEEE Trans on Pattern Analysis and Machine Intelligence 45:13567–13585

Reed S, Lee H, Anguelov D, et al (2015) Training deep neural networks on noisy labels with bootstrapping. In: ICLR

Ren M, Zeng W, Yang B, et al (2018) Learning to reweight examples for robust deep learning. In: ICML

Sharma K, Donmez P, Luo E, et al (2020) Noiserank: Unsupervised label noise reduction with dependence models. In: ECCV

Shen Y, Sanghavi S (2019) Learning with bad training data via iterative trimmed loss minimization. In: ICML

Shu J, Xie Q, Yi L, et al (2019) Meta-weight-net: Learning an explicit mapping for sample weighting. In: NeurIPS

Shu J, Yuan X, Meng D, et al (2023) Cmw-net: Learning a class-aware sample weighting mapping for robust deep learning. IEEE Trans on Pattern Analysis and Machine Intelligence 45(10):11521–11539

Sohn K, Berthelot D, Carlini N, et al (2020) Fixmatch: Simplifying semi-supervised learning with consistency and confidence. NeurIPS

Song H, Kim M, Lee JG (2019) Selfie: Refurbishing unclean samples for robust deep learning. In: ICML

Sukhbaatar S, Bruna J, Paluri M, et al (2015) Training convolutional networks with noisy labels. In: ICLR

Sun H, Guo C, Wei Q, et al (2022) Learning to rectify for robust learning with noisy labels. Pattern Recognition 124:108467

Tanno R, Saeedi A, Sankaranarayanan S, et al (2019) Learning from noisy labels by regularized estimation of annotator confusion. In: CVPR

Vahdat A (2017) Toward robustness against label noise in training deep discriminative neural networks. In: NeurIPS

Wang X, Kodirov E, Hua Y, et al (2019) Improving mae against cce under label noise. arXiv preprint arXiv:190312141

Wang Y, Kucukelbir A, Blei DM (2017) Robust probabilistic modeling with bayesian data reweighting. In: ICML

Wang Z, Hu G, Hu Q (2020) Training noise-robust deep neural networks via meta-learning. In: CVPR

Wei H, Feng L, Chen X, et al (2020) Combating noisy labels by agreement: A joint training method with co-regularization. In: CVPR

Wei Q, Sun H, Lu X, et al (2022) Self-filtering: A noise-aware sample selection for label noise with confidence penalization. In: ECCV

Wei Q, Feng L, Sun H, et al (2023) Fine-grained classification with noisy labels. In: CVPR

Wu Y, Shu J, Xie Q, et al (2021) Learning to purify noisy labels via meta soft label corrector. In: AAAI

Xia X, Liu T, Han B, et al (2020a) Robust early-learning: Hindering the memorization of noisy labels. In: ICLR

Xia X, Liu T, Han B, et al (2020b) Part-dependent label noise: Towards instance-dependent label noise. In: NeurIPS

Xia X, Han B, Zhan Y, et al (2023) Combating noisy labels with sample selection by mining high-discrepancy examples. In: ICCV

Xiao T, Xia T, Yang Y, et al (2015) Learning from massive noisy labeled data for image classification. In: CVPR

Xu Y, Zhu L, Jiang L, et al (2021a) Faster meta update strategy for noise-robust deep learning. In: CVPR

Xu Y, Zhu L, Jiang L, et al (2021b) Faster meta update strategy for noise-robust deep learning. In: CVPR

Yao Y, Liu T, Han B, et al (2020) Dual t: Reducing estimation error for transition matrix in label-noise learning. In: NeurIPS

Yao Y, Gong M, Du Y, et al (2023) Which is better for learning with noisy labels: The semi-supervised method or modeling label noise? In: ICML

Yu X, Han B, Yao J, et al (2019) How does disagreement help generalization against label corruption? In: ICML

Yu X, Jiang Y, Shi T, et al (2023) How to prevent the continuous damage of noises to model training? In: CVPR

Yuan S, Feng L, Liu T (2023) Late stopping: Avoiding confidently learning from mislabeled examples. In: ICCV

Zadrozny B (2004) Learning and evaluating classifiers under sample selection bias. In: ICML

Zagoruyko S, Komodakis N (2016) Wide residual networks. In: BMVC

Zhang H, Cisse M, Dauphin YN, et al (2018) mixup: Beyond empirical risk minimization. In: ICLR

Zhang W, Wang Y, Qiao Y (2019) Metacleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition. In: CVPR

Zhang Y, Niu G, Sugiyama M (2021a) Learning noise transition matrix from only noisy labels via total variation regularization. In: ICML

Zhang Y, Zheng S, Wu P, et al (2021b) Learning with feature-dependent label noise: A progressive approach. In: ICLR

Zhang Z, Sabuncu MR (2018) Generalized cross entropy loss for training deep neural networks with noisy labels. In: NeurIPS

Zhao Q, Shu J, Yuan X, et al (2023) A probabilistic formulation for meta-weight-net. IEEE Trans on Neural Networks and Learning Systems 34(3):1194–1208

Zheng G, Awadallah AH, Dumais S (2021) Meta label correction for noisy label learning. In: AAAI

Zhou X, Liu X, Wang C, et al (2021) Learning with noisy labels via sparse regularization. In: ICCV

Zhu Z, Liu T, Liu Y (2021) A second-order approach to learning with instance-dependent label noise. In: CVPR