Maximising the Utility of Validation Sets for Imbalanced Noisy-label Meta-learning

Hoang Anh Dung
University of Adelaide
a1745254@adelaide.edu.au

Cuong Nguyen
University of Adelaide
cuong.nguyen@adelaide.edu.au

Vasileios Belagiannis
Otto von Guericke University Magdeburg
vasileios.belagiannis@ovgu.de

Gustavo Carneiro
University of Adelaide
gustavo.carneiro@adelaide.edu.au

Abstract

Meta-learning is an effective method to handle imbalanced and noisy-label learning, but it depends on a validation set containing randomly selected, manually labelled and balanced distributed samples. The random selection and manual labelling and balancing of this validation set is not only sub-optimal for meta-learning, but it also scales poorly with the number of classes. Hence, recent meta-learning papers have proposed ad-hoc heuristics to automatically build and label this validation set, but these heuristics are still sub-optimal for meta-learning. In this paper, we analyse the meta-learning algorithm and propose new criteria to characterise the utility of the validation set, based on: 1) the informativeness of the validation set; 2) the class distribution balance of the set; and 3) the correctness of the labels of the set. Furthermore, we propose a new imbalanced noisy-label meta-learning (INOLML) algorithm that automatically builds a validation set by maximising its utility using the criteria above. Our method shows significant improvements over previous meta-learning approaches and sets the new state-of-the-art on several benchmarks.

1 Introduction

Within the past decade, there have been great advancements in visual classification [24, 19, 61], object detection [73, 51, 36] and segmentation [7, 32] thanks in part to deep learning models. The functionality of these models partly depends on large training sets containing samples that have been correctly labelled and that are well-balanced among the classes. The difficulty in obtaining such training sets is motivating researchers to develop methods that can work with less well curated data sets [65, 35]. Unfortunately, such poorly curated datasets are more likely to contain label noise and imbalanced class distribution.

In the literature, the problems of imbalanced learning and noisy-label learning are generally treated separately. While noisy label methods are based on robust loss functions [58, 62], label cleaning [22, 70], meta-learning [50, 18], ensemble learning [43], and other methods [28, 74], imbalanced learning approaches are based on meta-learning [50, 18, 79], transfer learning [10, 60], classifier design [64, 37], re-sampling [57], and etc. Among those approaches, meta-learning based methods [24, 50, 80, 79, 66, 52, 2, 1, 56] can address both noisy-label and imbalanced learning problems.

Meta-learning is often formulated as a bi-level optimisation, where the upper level estimates the meta parameters using the validation set, and the lower level trains a classifier using the training set and the estimated meta-parameters, where the validation set is commonly built by randomly selecting and manually labelling training samples. However, the process of building these validation sets scales poorly with the number of classes, and the random selection may not pick the most informative samples.

These issues have motivated the design of ad-hoc methods to build the validation set [79, 66]. Unfortunately, their results are not as competitive as approaches that rely on manually-curated validation sets. This issue may be due to a shortcoming in their proposed heuristics [79, 66], which characterises balanced distribution and label cleanliness but ignores the informativeness for the meta-learning algorithm.

In this paper, we propose a new imbalanced noisy-label meta-learning (INOLML) method that automatically builds a validation set by maximising its utility in terms of sample informativeness, class distribution balance, and label correctness. The central contribution of the paper is the definition of the validation set utility criteria, which is motivated by the bi-level optimisation meta-learning algorithm. The proposed method, depicted in Fig. 1, consists of an iterative 3-step approach, namely: 1) pseudo-clean sample detection and robust labelling from the noisy training set; 2) validation set formation from the robustly labelled pseudo-clean set in step (1), using the proposed utility criteria; and 3) meta learning using the validation set from step (2). The main contributions of our paper can be summarised as follows:

- A new method to build the meta-learning validation set by maximising its utility for sample informativeness, class distribution balance, and label correctness;
- An innovative meta-learning algorithm (Fig. 1), comprising the steps: 1) detection and robust labelling of pseudo-clean samples from the noisy training set; 2) formation of the validation set using the proposed utility criteria; and 3) meta learning using the validation set from step (2).

With the two technical contributions above, our method shows improvements over previous meta-learning approaches on imbalanced noisy-label learning benchmarks. In balanced noisy-label benchmarks, our method is competitive or better than the state-of-the-art.

2 Related Work

We review methods that can deal with imbalanced noisy-label learning, focusing on meta-learning approaches.

2.1 Noisy-label Learning

Current noisy-label learning methods can rely on many strategies, such as: robust loss functions [59, 62, 38], ensemble learning [43], student-teacher model [55], label cleaning [70, 22], co-teaching [33, 24, 41, 19, 68], dimensionality reduction [40], iterative label correction [77], semi-supervised learning [46, 33, 47], meta-learning [18, 53, 24, 50, 80, 79, 66, 52, 2, 1, 56], and hybrid methods [69, 28, 74, 45, 23]. Usually, most of the methods above assume that the training set has a balanced distribution of samples per class, except for the meta-learning approaches [24, 50, 80, 79, 66, 52, 2, 1, 56] that not only address the noisy-label problem, but also the learning with an imbalanced training set.

Meta learning is a versatile solution for many problems (few-shot learning, reinforcement learning, etc.) that optimises meta-parameters in order to benefit the training process. In noisy-label meta learning papers [24, 50, 80, 79, 66, 52, 2, 1, 56], the meta parameters consist of a weight for each training sample [79, 80], and the meta learning methods optimise the model based on a weighted cross entropy loss that automatically down-weights noisy samples and upweights clean samples. For example, L2LWS [13] and CWS [12] comprise a target deep neural network (DNN) and a meta-DNN that is pre-trained on a small clean validation dataset to re-weight the training samples to model the target DNN. Automatic reweighting [50] weights training samples based on the performance of one-step-ahead model on the validation set. Except for recent methods [79, 66], meta-learning approaches require a clean validation set that can be expensive to acquire or unavailable in real world scenarios. Therefore, similarly to [79, 66], we focus on the development of an approach that can automatically build a clean validation set, but unlike them, we propose an approach that is motivated by the meta-learning algorithm.

2.2 Imbalanced Learning

Imbalance learning is another challenging classification problem that is commonly present in real-world datasets, where a small portion of majority classes have a massive amount of training samples, and minority classes only have a few training samples [76]. This can easily result in a biased model that shows good accuracy

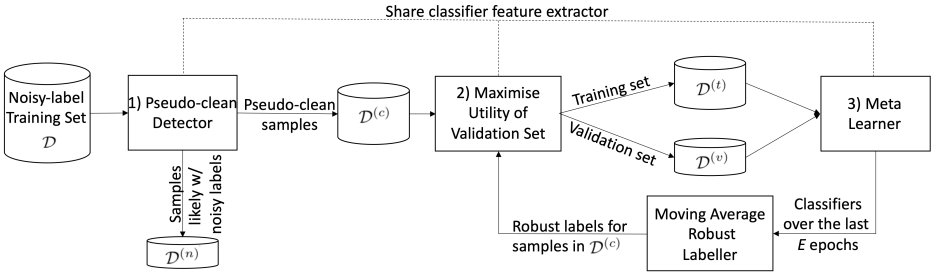


Figure 1: Main stages of INOLML: 1) classify the noisy-label samples from $\mathcal{D}$ into $\mathcal{D}^{(c)}$ (with samples that are likely to have clean labels) and $\mathcal{D}^{(n)}$ (samples likely to have noisy labels); 2) build a validation set $\mathcal{D}^{(v)}$ containing samples that are informative (from a meta-learning perspective), balanced and with a high likelihood of containing clean labels tested with the moving average robust labeller, where the training set $\mathcal{D}^{(t)} = \mathcal{D}^{(c)} \setminus \mathcal{D}^{(v)}$; and 3) train the meta-learning classifier with $\mathcal{D}^{(t)}$ and $\mathcal{D}^{(v)}$. These three steps are iterated during training.

for majority classes, but poor performance for the minority ones. To address this problem, many imbalanced learning methods have been proposed [60, 10, 37, 64, 57], where the main techniques are [76]: transfer learning [10, 60], classifier design [37, 64], re-sampling (e.g., meta-learning) [57], decoupled training [26, 25], ensemble learning [81, 17], cost-sensitive learning [82, 54, 15], data augmentation [71, 9], logit adjustment [42, 49] and representation learning [75, 21]. Unfortunately, existing methods designed to learn from long-tailed class distributions assume the labels to be clean, making their performance unclear in a more realistic scenario where the datasets are also noisy.

2.3 Noisy-label and Imbalanced Learning

Most of the papers listed in Sections 2.1 and 2.2 studied noisy label and imbalance learning problems as separate problems, except FSR [79] – a recent meta-learning approach that aims to solve both problems with meta-learning. The presence of label noise in imbalanced datasets has also been considered by non meta-learning approaches [4, 63, 27], but they either have different setups or achieve inferior results compared with recently proposed meta-learning approaches. As mentioned in Sec. 1, the validation set plays a central role in meta-learning, but we are not aware of papers that study how to maximise its utility during optimisation. Classic meta-learning approaches [50, 52] relies on a random sample selection and manually labelling approach that will likely result in a sub-optimal validation set. In addition, the fact that such manually-curated validation set is fixed for the whole training process may hinder the generalisation of the model. Recent meta-learning approaches try to automatically build a validation set that varies throughout the training process. For instance, FaMUS [66] selects training samples with low losses to form the validation set, while FSR [79] chooses samples that can be well optimized after a training iteration to build the validation set. Such methods, however, form a validation set based on heuristics that are not directly related to the meta-learning optimisation, which is the problem being studied in this paper.

3 Method

The initial training set is defined as $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{|\mathcal{D}|}$, with $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^{H \times W \times R}$ representing an image of size $H \times W$ pixels and R colour channels, and $\mathbf{y}_i \in \mathcal{Y} = \{\mathbf{v} : \mathbf{v} \in \{0,1\}^C \text{ and } \sum_{k=1}^{C} \mathbf{v}(k) = 1\}$ being the noisy one-hot and C denoting the number of classes [20]. The classification model is represented by $f_\theta : \mathcal{X} \to \Delta_{C-1}$ parameterised by $\theta \in \Theta$, with the $C-1$ probability simplex $\Delta_{C-1} = \{\mathbf{p} : \mathbf{p} \in [0,1]^C \text{ and } \sum_{k=1}^{C} \mathbf{p}(k) = 1\}$.

The proposed INOLML follows a bi-level optimisation [50, 80] that relies on the meta-parameter $\omega = \{\omega_i\}_{i=1}^{|\mathcal{D}^{(t)}|}$ ($\omega_i \geq 0$) that weights the samples in the

training set $\mathcal{D}^{(t)}$ based on their utility regarding informativeness and label cleanliness, and $\lambda = \{\lambda_i\}_{i=1}^{|\mathcal{D}^{(t)}|}$ ($\lambda_i \in [0,1]$) that weights the contribution of model prediction in the pseudo-label estimation, as in $\hat{\mathbf{y}}_i(\lambda_i) = \lambda_i \mathbf{y}_i + (1-\lambda_i) f_\theta(\mathbf{x}_i)$. The meta-learning optimisation is defined by:

$$\omega^*, \lambda^* = \arg\min_{\omega,\lambda} \frac{1}{|\mathcal{D}^{(v)}|} \sum_{(\mathbf{x}_j,\mathbf{y}_j)\in\mathcal{D}^{(v)}} \ell^{(v)}\left(\mathbf{x}_j, \mathbf{y}_j; \theta^*(\omega,\lambda)\right)$$
$$\text{s.t.: } \theta^*(\omega,\lambda) = \arg\min_{\theta} \frac{1}{|\mathcal{D}^{(t)}|} \sum_{(\mathbf{x}_i,\mathbf{y}_i)\in\mathcal{D}^{(t)}} \omega_i\, \ell^{(t)}\left(\mathbf{x}_i, \hat{\mathbf{y}}(\lambda_i); \theta\right), \tag{1}$$

where: $\ell^{(v)}(\mathbf{x}_j, \mathbf{y}_j; \theta^*(\omega,\lambda)) = \ell_{\text{CE}}(\mathbf{y}_j, f_{\theta^*(\omega,\lambda)}(\mathbf{x}_j))$ is the cross-entropy (CE) loss between the label $\mathbf{y}_j$ and model prediction, $\ell^{(t)}\left(\mathbf{x}_i, \hat{\mathbf{y}}(\lambda_i); \theta\right)$ is defined below in (12). The validation set $\mathcal{D}^{(v)}$ is obtained by:

$$\mathcal{D}^{(v)} = \mathsf{MaxUtility}\left(\mathcal{D}^{(c)}\right), \tag{2}$$

which depends on the pseudo-clean set $\mathcal{D}^{(c)}$ obtained from:

$$\mathcal{D}^{(c)} = \mathsf{PseudoCleanDetector}\left(\mathcal{D}\right). \tag{3}$$

Our main contribution is the definition of the utility criteria in $\mathsf{MaxUtility}(\mathcal{D}^{(c)})$ in (2) to select the validation set $\mathcal{D}^{(v)}$ and training set $\mathcal{D}^{(t)}$, where $\mathcal{D}^{(v)} \subset \mathcal{D}^{(c)}$, with $\mathcal{D}^{(t)} \cap \mathcal{D}^{(v)} = \varnothing$ and $\mathcal{D}^{(t)} \cup \mathcal{D}^{(v)} = \mathcal{D}$. The validation set $\mathcal{D}^{(v)}$ has a balanced distribution of samples per class, contains samples that are informative for the meta-learning optimisation in (1) and are likely to have clean labels.

The pseudo-clean sample set $\mathcal{D}^{(c)}$ is estimated with a noisy-label classifier, represented by $\mathsf{PseudoCleanDetector}(.)$ shown in (3). Such classifier selects pseudo-clean samples based on the small CE loss hypothesis [19, 33] where the loss is computed between the labels in $\mathcal{D}$ and the predictions by $f_\theta(.)$. The remaining samples form $\mathcal{D}^{(n)}$, with $\mathcal{D}^{(c)} \cup \mathcal{D}^{(n)} = \mathcal{D}$ and $\mathcal{D}^{(c)} \cap \mathcal{D}^{(n)} = \varnothing$. These sets are regularly updated during training. The initial pseudo-clean set at the first training iteration is estimated from the model $f_\theta(.)$ trained with early-stopping. In the following subsections, we describe how to select a validation set that maximises its utility in terms of informativeness, label cleanliness and class distribution balance.

3.1 Maximising the Utility of the Validation Set

The maximisation of the utility of the validation set is motivated by the bi-level optimisation in (1), where we focus on the weighting of each training sample, represented by ω_i, which estimates the importance of that sample in the training process. The optimisation in (1) is solved by iterating the following 2 steps. In the first step, the locally-optimal model parameter $\theta^*(\omega,\lambda)$ in the lower-level is obtained by applying stochastic gradient descent (SGD) on the training set $\mathcal{D}^{(t)}$ with each step defined by:

$$\hat{\theta}(\omega,\lambda) = \theta(\omega,\lambda) - \eta_\theta \nabla_\theta \left(\frac{1}{|\mathcal{D}^{(t)}|} \sum_{(\mathbf{x}_i,\mathbf{y}_i)\in\mathcal{D}^{(t)}} \omega_i\, \ell^{(t)}(\mathbf{x}_i, \hat{\mathbf{y}}_i(\lambda_i); \theta) \right)\Bigg|_{\theta=\theta(\omega,\lambda)}. \tag{4}$$

In the second step, the meta-parameters, ω and λ, in the upper-level, are updated by applying one SGD step on the validation set $\mathcal{D}^{(v)}$. For ω, the update is defined as:

$$\omega_i^* = \max\left(0, -\frac{\eta_\omega}{|\mathcal{D}^{(v)}|} \sum_{(\mathbf{x}_j,\mathbf{y}_j)\in\mathcal{D}^{(v)}} \frac{\partial}{\partial \omega_i} \ell^{(v)}(\mathbf{x}_j, \mathbf{y}_j; \theta^*(\omega,\lambda))\Big|_{\omega_i=0}\right), \tag{5}$$

and the update for λ is defined below in (11). The obtained meta-parameters are then used in the next bi-level optimisation iteration.

According to [50], the gradient w.r.t. ω is expressed as:

$$\sum_{(\mathbf{x}_j,\mathbf{y}_j)\in\mathcal{D}^{(v)}} \frac{\partial}{\partial \omega_i} \ell^{(v)}(\mathbf{x}_j, \mathbf{y}_j; \theta^*(\omega,\lambda))\Big|_{\omega_i=0} \propto - \sum_{(\mathbf{x}_j,\mathbf{y}_j)\in\mathcal{D}^{(v)}} \sum_{l=1}^{L} (\mathbf{z}_{j,l-1}^{(v)\top} \mathbf{z}_{i,l-1}^{(t)})(\mathbf{g}_{j,l}^{(v)\top} \mathbf{g}_{i,l}^{(t)}), \tag{6}$$

where $\mathbf{z}_{j,l-1}^{(v)}$ denotes the feature from validation image $\mathbf{x}_j$ to be processed by layer l of the model (similarly for the training image feature $\mathbf{z}_{i,l-1}^{(t)}$), and $\mathbf{g}_{j,l}^{(v)}$ represents gradient from layer l for the validation image $\mathbf{x}_j$ (similarly for the training image gradient $\mathbf{g}_{i,l}^{(t)}$). Hence, the weight of a training sample is high if both its feature and gradient are similar to the feature and gradient of at least one of the validation samples; otherwise, the weight is low.

Therefore, a validation set that maximises the weight of samples in the training set maximises its utility for

the meta-learning optimisation. This observation is at the crux of our validation sample selection approach, where we first form a pseudo-clean set from the training set and then search within that pseudo-clean set to form a validation set that is balanced and maximises the sum in (6). The validation set $\mathcal{D}^{(v)} \subset \mathcal{D}^{(c)}$ is built with the function $\mathsf{MaxUtility}(.)$ from (2) with the following bi-level optimisation:

$$\mathcal{D}^{(v)} = \arg \max_{\substack{\widehat{\mathcal{D}}^{(v)} \subset \widetilde{\mathcal{D}}^{(v)} \\ |\widehat{\mathcal{D}}^{(v)}| = M \times C}} \mathsf{Clean}\left(\widehat{\mathcal{D}}^{(v)}, \mathcal{D}^{(c)}\right)$$
$$\text{s.t.: } \widetilde{\mathcal{D}}^{(v)} = \arg \max_{\substack{\bar{\mathcal{D}}^{(v)} \subset \mathcal{D}^{(c)} \\ |\bar{\mathcal{D}}^{(v)}| = K \times C}} \mathsf{Info}\left(\bar{\mathcal{D}}^{(v)}, \mathcal{D}^{(c)}\right). \quad (7)$$

The function $\mathsf{Info}(.)$ in the lower-level of (7) is defined as:

$$\mathsf{Info}(\bar{\mathcal{D}}^{(v)}, \mathcal{D}^{(c)}) = \sum_{(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}^{(c)} \setminus \bar{\mathcal{D}}^{(v)}} \max_{\substack{(\mathbf{x}_j, \mathbf{y}_j) \in \bar{\mathcal{D}}^{(v)} \\ \mathbf{y}_j = \mathbf{y}_i}} \iota(\mathbf{x}_i, \mathbf{x}_j), \quad (8)$$

with

$$\iota(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^{L} (\mathbf{z}_{j,l-1}^\top \mathbf{z}_{i,l-1})(\mathbf{g}_{j,l}^\top \mathbf{g}_{i,l}), \quad (9)$$

where, similarly to (6), $\mathbf{z}_{j,l-1}$ is the image feature input to layer l from $\mathbf{x}_j$ (same for $\mathbf{z}_{i,l-1}$ from $\mathbf{x}_i$), and $\mathbf{g}_{j,l}$ denotes the validation image gradient of layer l from $\mathbf{x}_j$ (same for $\mathbf{g}_{i,l}$ from $\mathbf{x}_i$). Note that the function $\iota(.)$ defined in (9) is the weight defined in (6) between the training sample $(\mathbf{x}_i, \mathbf{y}_i)$ and the validation sample $(\mathbf{x}_j, \mathbf{y}_j)$, or the "information" that $(\mathbf{x}_i, \mathbf{y}_i)$ can get from $(\mathbf{x}_j, \mathbf{y}_j)$. Intuitively, the lower-level summation in (7) (and in particular (8)) is designed to form a candidate balanced set $\widetilde{\mathcal{D}}^{(v)}$ by maximising the maximum "information content" that the pseudo-clean samples from $\mathcal{D}^{(c)} \setminus \widetilde{\mathcal{D}}^{(v)}$ can get from the samples in $\widetilde{\mathcal{D}}^{(v)}$. The reason we maximise the maximum instead of the average "information content" is to guarantee that each clean training sample get upweighted by at least one clean validation sample. Unfortunately, the samples selected to be in $\widetilde{\mathcal{D}}^{(v)}$ can still have noisy labels since $D^{(c)}$ is not completely clean and the function $\mathsf{Info}(.)$ tends to return high values if samples in $\widetilde{\mathcal{D}}^{(v)}$ have low confidence logit scores and high gradient values. Simply filtering out samples with higher gradient will force the validation set to contain samples that are more likely to be clean, but less likely to be informative. Therefore, we aim to identify samples that are likely to have clean labels without relying on their prediction logits.

To search for clean samples in $\widetilde{D}^{(v)}$, we observe that the samples from this set are more likely to be clean when they have higher similarity with other samples belonging to the same class. Given this observation, we therefore propose a heuristic based on the cosine similarity between the sample of interest and other samples of the same class in the pseudo-clean set $\mathcal{D}^{(c)}$. The heuristic is represented by the function $\mathsf{Clean}(.)$, which is defined as follows:

$$\mathsf{Clean}\left(\widehat{\mathcal{D}}^{(v)}, \mathcal{D}^{(c)}\right) = \sum_{\substack{(\mathbf{x}_j, \mathbf{y}_j) \in \widehat{\mathcal{D}}^{(v)} \\ (\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{D}^{(c)} \setminus \widehat{\mathcal{D}}^{(v)} \\ \mathbf{y}_i = \mathbf{y}_j}} \sum_{l=1}^{L} \left(\mathbf{z}_{j,l-1}^\top \mathbf{z}_{i,l-1}\right). \quad (10)$$

We also impose a constraint that selects M samples for each of the C classes with $M \ll K$ as shown in the upper-level of (7) to obtain a balanced validation subset $\mathcal{D}^{(v)}$.

Given that both optimisations in (7) consist of combinatorial problems, we resort to a greedy approach that loops through the classes and sequentially selects M and K samples for each class (for the upper and lower optimisation, respectively) that maximise the respective objective function. As the solution needs to iterate through all layers of a neural network of interest, the calculation of gradient in (6) and the optimisation in (7) might be expensive, especially for large-scale deep neural networks. However, according to Zhang et al. [79], the weights of training samples in meta-learning mostly depend on the last layer of the model. Hence, we use only on the last layer L of the model for (6) and (7) to reduce the computational cost.

3.2 Training Procedure

Our training procedure follows the 3-step iterative approach depicted in Fig. 1, where step 1 (pseudo-clean label detector) and step 2 (maximise the utility of the validation set) have been explained in Section 3.1. Step 3 (meta-learning) is based on the optimisation in (1), where our training loss $\ell(.)$ follows the one defined in [80].

To optimise (1), we first estimate ω^* with (5) and λ^* (i.e., the pseudo-labelling parameter defined in (1))

with [80]:

$$\lambda_i^* = \left[\text{sign} \left(\sum_{(\mathbf{x}_j, \mathbf{y}_j) \in \mathcal{D}^{(v)}} \frac{\partial}{\partial \lambda_i} \ell^{(v)}(\mathbf{x}_j, \mathbf{y}_j; \theta^*(\omega, \lambda)) \right) \right]_+, \tag{11}$$

where $\ell^{(v)}(\mathbf{x}_j, \mathbf{y}_j; \theta^*(\omega, \lambda))$ is defined in (1).

After estimating ω^* and λ^*, we optimise the lower level of (1) to estimate the model parameter using the following loss function [80]:

$$\begin{aligned} \ell^{(t)}(\mathbf{x}_i, \mathbf{y}_i; \theta) =& \omega_i^* \ell_{\text{CE}}(\widehat{\mathbf{y}}_i(\lambda_0), f_\theta(\mathbf{x}_i)) + \frac{1}{B} \ell_{\text{CE}}(\mathbf{y}_i^*(\lambda_i^*), f_\theta(\mathbf{x}_i)) + \\ & p \times \ell_{\text{CE}}(\mathbf{y}_i^\beta, f_\theta(\mathbf{x}_i^\beta)) + k \times \ell_{\text{KL}}(f_\theta(\mathbf{x}_i), f_\theta(\hat{\mathbf{x}}_i)), \end{aligned} \tag{12}$$

where $\widehat{\mathbf{y}}_i(\lambda_0)$ is a pseudo-label, defined as in (1), with a fixed weight $\lambda_0 = 0.9$, $\mathbf{y}_i^*(\lambda_i^*) = \mathbf{y}_i$ if $\lambda_i^* > 0$, $\mathbf{y}_i^* = f_\theta(\mathbf{x}_i)$ if $\lambda_i^* \leq 0$, $\mathbf{y}_i^\beta$ and $\mathbf{x}_i^\beta$ are obtained via the mixup operator [72] using the training and validation sets, $\ell_{\text{KL}}(.,.)$ represents the Kullback-Leibler (KL) divergence [30] between the model response for training image $\mathbf{x}_i$ and its data augmented version $\hat{\mathbf{x}}_i$, p and k are hyper-parameters, and B is the batch size.

The effectiveness of the optimisation in (7) depends on the actual (hidden) proportion of clean samples in the pseudo clean set $D^{(c)}$, while the efficiency depends on the size of $D^{(c)}$. Hence, to reduce computational cost, the selection of the validation set in (7) uses a subset $\widetilde{D}^{(c)} \subset \{(\mathbf{x}_i, \mathbf{y}_i) : (\mathbf{x}_i, \mathbf{y}_i) \in D^{(c)} \wedge \arg\max_{k \in \{1,\dots,C\}} \mathbf{y}_i(k) = \arg\max_{k \in \{1,\dots,C\}} \tilde{\mathbf{y}}_i(k)\}$. This subset contains N randomly-selected samples $(\mathbf{x}_i, \mathbf{y}_i)$ of each class in $D^{(c)}$ that have their observed labels $\mathbf{y}_i$ consistent with the corresponding moving average robust label computed with the average prediction over the last E epochs, as in $\tilde{\mathbf{y}}_i = \kappa \tilde{\mathbf{y}}_i + {}^{(1-\kappa)}/_E \sum_{e=1}^{E} f_\theta(\mathbf{x}_i)$, with $\kappa \in [0, 1]$ being a hyper-parameter. The details of the training process are in Algorithm 1 of Appendix A.

4 Experiment and Analysis

We evaluate our method INOLML on four datasets: CIFAR10, CIFAR100 [29], WebVision [35] and Controlled Noisy Web Labels (CNWL) [66] with different noise settings, including symmetric, asymmetric, openset [3, 61], and long-tailed imbalance with and without symmetric noise [79]. For each type of experiment, we keep the noisy training set the same across all models for a fair comparison.

4.1 Datasets

CIFAR10 and CIFAR100 datasets [29] contain 50k and 10k images used for training and testing, respectively. Each image has size 32×32 pixels and is labelled as one of 10 or 100 classes. WebVision [35] is a dataset of 2.4 million images crawled from Google and Flickr based on the 1,000 ImageNet classes [14]. The dataset is more challenging than CIFAR since it is class-imbalanced and contains real-world noisy labels. Following [79], we extract a subset that contains the first 50 classes to create the WebVision mini dataset [24]. CNWL [23] is a new benchmark of controlled real-world label noise from the web that contains various noise rates ranging from 0 to 0.8. Following FaMUS [66], we evaluate the proposed method on Red Mini-ImageNet dataset that consists of 50k training images from 100 classes for training and 5k images for testing.

4.2 Implementation Details

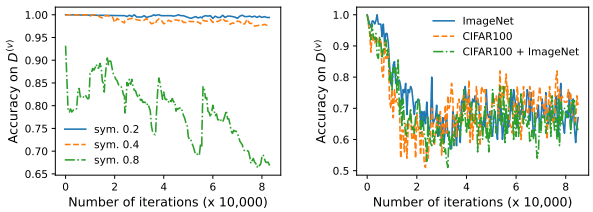
For all experiments on CIFAR datasets, except long-tail imbalance, we use the same hyperparameters and network architectures as the Distill model [80]. We adopt the cosine learning rate decay with warm restarting [39] and SGD optimiser. For CIFAR datasets, we train WideResnet28-10 with 100k iterations and a batch size of 100. We also train a smaller network (Resnet29) to fairly compare with [80]. For WebVision, we follow FSR [79] and train a single Resnet50 network with 1 million iterations and a batch size of 16. For Red mini-ImageNet, we run experiments with 150k iterations and a batch size of 100. For CNWL, we use a single PreAct Resnet18 network that is similar to previous works [11, 47] on this benchmark. For the class imbalance problems, we use the popular Resnet32 model to fairly compare with FaMUS [66] and FSR [79]. We report the prediction accuracy of each experiment on their corresponding testing sets. Please refer to Appendix B for implementation details and hyper-parameters values.

Table 1: Test accuracy (in %) of our INOLML and previous methods evaluated on various symmetric noise rates. Methods with superscript $^{\mathrm{T}}$ represent meta-learning methods that need clean validation sets. The lower block contains meta-learning methods while the upper block shows methods with SOTA results.

Method	CIFAR10			CIFAR100		
	0.2	**0.4**	**0.8**	**0.2**	**0.4**	**0.8**
GJS[78]	95.3 ± 0.2	93.6 ± 0.2	79.1 ± 0.3	78.1 ± 0.3	75.7 ± 0.3	44.5 ± 0.5
DivideMix[31]	95.7 ± 0.0	-	92.9 ± 0.0	76.9 ± 0.0	-	59.6 ± 0.0
CRUST[44]	91.1 ± 0.2	89.2 ± 0.2	58.3 ± 1.8	-	-	-
PENCIL[67]	-	-	-	73.9 ± 0.3	69.1 ± 0.6	-
ELR[38]	92.1 ± 0.4	91.4 ± 0.2	80.7 ± 0.6	74.7 ± 0.3	68.4 ± 0.4	30.2 ± 0.8
FaMUS [78]	-	95.3 ± 0.2	-	-	76.0 ± 0.2	-
Distill$^{\mathrm{T}}$ [80]	96.2 ± 0.2	95.9 ± 0.2	93.7 ± 0.5	81.2 ± 0.7	80.2 ± 0.3	**75.5 ± 0.2**
MentorNet$^{\mathrm{T}}$[24]	92.0 ± 0.0	89.0 ± 0.0	49.0 ± 0.0	73.0 ± 0.0	68.0 ± 0.0	35.0 ± 0.0
L2R$^{\mathrm{T}}$[50]	90.0 ± 0.4	86.9 ± 0.2	73.0 ± 0.8	67.1 ± 0.1	61.3 ± 2.0	35.1 ± 1.2
MWN$^{\mathrm{T}}$ [52]	90.3 ± 0.6	87.5 ± 0.2	-	64.2 ± 0.3	58.6 ± 0.5	-
GDW$^{\mathrm{T}}$ [6]	-	88.1 ± 0.4	-	-	59.8 ±1.6	-
FSR[79]	95.1 ± 0.1	93.7 ± 0.1	82.8 ± 0.3	78.7 ± 0.2	74.2 ± 0.4	46.7 ± 0.8
INOLML	**96.9 ± 0.1**	**96.6 ± 0.1**	**95.0 ± 0.2**	**82.0 ± 0.2**	**81.3 ± 0.2**	74.7 ± 0.1

4.3 Symmetric Noise

Table 1 shows the test accuracy of many methods, including meta-learning based approaches that require a clean validation set (indicated with $^{\mathrm{T}}$) and others that automatically build their validation sets, at various level of noise rates ranging from 20% to 80%. In general, the proposed method outperforms most of the previous methods, even though we do not require a clean validation set. The slightly lower performance than Distill on CIFAR100 at 80% noise rate can be explained by Distill's large manually curated clean validation set with 10 clean samples per class. In addition, as shown in Fig. 2a, at 80% symmetric noise rate, a significant portion (20% to 45%) of our clean validation set contains noisy samples at the final training stages, which deteriorates the efficacy of our approach. We also carry out additional experiments with different validation set sizes to fairly compare with Distill in Appendix C, in which our method outperforms Distill by 1 to 3% in majority of scenarios. Overall, these results show that a pseudo-clean, balanced, and informative validation set, can outperform a randomly-collected clean validation set in most symmetric noise scenarios. Our results also set new state-of-the-art (SOTA) results on the symmetric label noise benchmarks for methods that do not require clean validation set.



(a) Symmetric noise benchmarks (b) Open-set noise benchmarks

Figure 2: Accuracy of the clean validation set $\mathcal{D}^{(v)}$ as training progresses evaluated on different noise benchmarks.

4.4 Asymmetric Noise

We compare our algorithm with Distill [80], FSR [79] and other approaches on CIFAR10 at 40% asymmetric noise rate. Similarly to the symmetric noise cases, we also use (pseudo-)clean validation sets of sizes 1, 5 and 10 samples per class and show the results of Distill and our method in Table 2 *(table on left)*. Although our proposed method does not rely on a manually-labelled validation set, it performs better than Distill, especially with small model architectures (Resnet29) and small validation sets (1 sample per class). Our active selection strategy has slightly lower accuracy with larger clean validation set sizes (larger than or equal 5 random clean samples per classes) on larger model architectures (WideResnet28). This might be caused by the confirmation bias of asymmetric noise in our selected pseudo-clean validation subset and the high capacity of larger models, such as WideResnet28-10, which are more prone to overfit label noise, especially when being trained on a small dataset with only 10 classes. We further evaluate the proposed method and show the higher performance of our method compared to other methods, such as FSR and L2R meta-learning methods, in Table 2 *(table on right)*.

4.5 Imbalanced Learning

We evaluate our INOLML on the imbalanced (long-tailed) CIFAR datasets following the same setting as [79]. The prediction accuracy in Table 3 shows that INOLML considerably surpasses all previous meta-learning ap-

Table 2: Test accuracy (in %) of our INOLML and previous methods on CIFAR10 with 0.4 asymmetric noise. *(table on left):* comparison with Distill using a validation set $\mathcal{D}^{(v)}$ of sizes 1, 5 and 10 samples per class on Resnet29 and WideResnet28-10, and *(table on right):* comparison with some leading methods. The superscript $^{\text{T}}$ indicates the need for clean validation sets.

Method	$\lvert\mathcal{D}^{(v)}\rvert$	Resnet29	WRN28-10
Distill$^{\text{T}}$	$1\times C$	76.8 ± 2.9	93.2 ± 0.2
INOLML		86.8 ± 0.9	93.6 ± 0.3
Distill$^{\text{T}}$	$5\times C$	86.7 ± 0.5	94.5 ± 0.2
INOLML		89.3 ± 0.2	94.1 ± 0.1
Distill$^{\text{T}}$	$10\times C$	88.6 ± 0.7	$\mathbf{94.9 \pm 0.1}$
INOLML		$\mathbf{89.8 \pm 0.3}$	94.2 ± 0.1

Method	Accuracy
GJS[16]	89.7 ± 0.4
F-Correction [48]	83.6 ± 0.3
PENCIL[67]	91.2 ± 0.0
DivideMix [33]	92.1 ± 0.0
MLNT [34]	92.3 ± 0.1
L2R$^{\text{T}}$[50]	90.8 ± 0.3
FSR[79]	93.6 ± 0.3
INOLML	$\mathbf{94.2 \pm 0.1}$

Table 3: Test accuracy (in %) of our INOLML and other SOTA meta-learning approaches evaluated on the CIFAR imbalanced learning (long-tailed) recognition task. The reported results are from Zhang et al. [79] and Xu et al. [66].

	CIFAR10			CIFAR100		
Imb. ratio	**200**	**50**	**10**	**200**	**50**	**10**
Softmax [79]	65.68	74.81	86.39	34.84	43.85	55.71
CB-Focal [79]	65.29	76.71	86.66	32.62	44.32	55.78
CB-Best [79]	68.89	79.27	87.49	36.23	45.32	57.99
L2R [50]	66.51	78.93	85.19	33.38	44.44	53.73
MWN [52]	68.91	80.06	87.84	37.91	46.74	58.46
GDW [6]	-	-	86.8	-	-	56.8
FaMUS [66]	-	83.32	87.90	-	49.93	59.03
FSR-DF [79]	66.15	79.78	88.15	36.74	44.43	55.60
FSR [79]	67.76	79.17	87.40	35.44	42.57	55.45
INOLML	**74.91**	**84.43**	**90.81**	**41.52**	**51.35**	**62.07**

proaches.

4.6 Imbalanced Noisy-label Learning

We evaluate our proposed method in the setting that combines class imbalance and label noise, which has been proposed in [79]. We follow the same experimental configuration by adding 20% and 40% symmetric noise to the CIFAR10 imbalanced datasets with different imbalance ratios (10, 50 and 200). The results in Table 4 show that our proposed method outperforms the benchmarks by a large margin. This result is even more remarkable when studying the results with a noise rate of 40%. For CIFAR100, we show the results for 20% and 40% symmetric noise and imbalance ratio 10. We do not show results for larger imbalance ratios because it was not possible to build validation sets with 10 samples per class for the minority classes. Nevertheless, for the two CIFAR100 problems, our method shows substantially better results than previous SOTA methods. Our method can therefore be considered the new SOTA in this imbalanced noisy-label learning benchmark with Resnet32 model.

Table 4: Test accuracy (in %) of our INOLML and other SOTA methods on CIFAR10 long-tailed recognition mixed with symmetric noise. The reported results are collected from [79] and [63].

Dataset	**Cifar10**						**Cifar100**	
Noise ratio	**0.2**			**0.4**			**0.2**	**0.4**
Imb. ratio	**10**	**50**	**200**	**10**	**50**	**200**	**10**	**10**
CRUST[44]	65.7	41.5	34.3	59.5	32.4	28.8	-	-
LDAM[5]	82.4	-	-	71.4	-	-	48.1	36.7
LDAM-DRW[5]	83.7	-	-	74.9	-	-	50.4	39.4
BBN[81]	80.4	-	-	70.0	-	-	47.9	35.2
HAR-DRW[4]	82.4	-	-	77.4	-	-	46.2	37.4
ROLT-DRW[63]	85.5	-	-	82.0	-	-	52.4	46.3
FSR[79]	85.7	77.4	65.5	81.6	69.8	49.5	-	-
INOLML	**90.1**	**80.1**	**66.6**	**89.1**	**78.1**	**61.6**	**59.8**	**56.1**

4.7 Open-set Noise

This type of noise refers to training images that belong to classes falling outside the C classes in $\mathcal{D}$. We follow [31], which forms 3 benchmarks using CIFAR10 contaminated with images from CIFAR100 and ImageNet. We compare with Distill and other meta-learning methods [78, 48, 67, 50, 79] in Table 5, and our method shows significant improvements in all benchmarks. In particular, comparing to Distill, our method is 0.5% to 1% better. One interesting observation is that our method outperforms Distill in the open-set noise even though the selected validation set $\mathcal{D}^{(v)}$ is largely contaminated with noisy samples (up to 40%) as shown in Fig. 2b. This is in contrast to our previous observation in symmetric and asymmetric noise settings where the more noisy samples in $\mathcal{D}^{(v)}$, the worse performance of the models trained with our method compared to Distill. Such difference might be attributed to the out-of-distribution characteristic of open-set noise. As open-set noisy-label datasets contain samples that do not belong to the set of known classes, such samples might help to regularise the learning on mislabelled data, reduc-

Table 5: Test accuracy (in %) of our INOLML and previous methods in open-set noise [3] using WideResnet28-10 with 10 samples per class for the validation set.

Method	ImageNet	CIFAR100	BOTH
RoG [48]	83.4	87.1	84.4
L2R [50]	81.8	81.8	85.0
Distill [80]	94.0	92.3	93.0
INOLML	$\mathbf{94.5 \pm 0.1}$	$\mathbf{93.6 \pm 0.0}$	$\mathbf{93.6 \pm 0.1}$

ing the effect of confirmation bias, resulting in a better performance.

4.8 Real-world Datasets

Table 6 shows the results of our method and other SOTA approaches on real-world datasets. Except for HAR[4] that uses InceptionResnetV2, Table 6 *(upper table)* shows the performance on WebVision with Resnet50, while Table 6 *(lower table)* shows results on four different noise ratios evaluated on Red Mini-ImageNet. In general, our method outperforms many SOTA methods on WebVision and is competitive with the best method [47] on Red Mini-ImageNet. We note that our proposed method is more efficient in terms of memory footprint than most of Co-training based approaches [33, 11, 66] evaluated on Red Mini-ImageNet since we use only a single PreAct Resnet18 model with meta-learning instead of two separate PreAct Resnet18 models.

Table 6: SOTA prediction accuracy (in %) comparison on real-world datasets. *(upper table):* WebVision mini dataset (50 classes) using Resnet50 evaluated on Webvision and ImageNet test set; and *(lower table):* Red Mini-ImageNet. The results of other methods are reported from Zhang et al.[79], PropMix[11] and their own works.

Method	WebVision		ImageNet	
	top-1	top-5	top-1	top-5
HAR [4]	75.5	90.7	57.4	82.4
D2L [40]	62.7	84.0	57.8	81.4
Co-teaching [19]	63.6	85.2	61.5	84.7
Iterative-CV [8]	65.2	85.3	61.6	85.0
MentorNet [24]	63.0	81.4	63.8	85.8
CRUST [44]	72.4	89.6	67.4	87.8
FSR [79]	74.9	88.2	72.3	87.2
GJS [16]	78.0	90.6	74.4	91.2
MW-Net [52]	74.5	88.9	72.6	88.8
INOLML	**81.7**	**93.8**	**78.1**	**92.9**

Method	Noise ratio			
	0.2	0.4	0.6	0.8
Cross entropy [11]	47.36	42.70	37.30	29.76
Mix Up [11]	49.10	46.40	40.58	33.58
DivideMix [33]	50.96	46.72	43.14	34.50
MentorMix[23]	51.02	47.14	43.80	33.46
FaMUS[66]	51.42	48.06	45.10	35.50
PropMix[11]	61.24	56.22	52.84	43.42
MOIT [47]	63.14	**60.78**	-	**45.88**
INOLML	**63.23**	58.21	**53.39**	**45.32**

5 Ablation Study and Discussion

We first study the optimisation in Eq. (7). In the lower-lever optimisation of Eq. (7), the function $\mathsf{Info}(.)$ not just select samples that maximise the training sample weight from Eq. (6), as that may lead to scenarios where most of selected samples are located in the same region of the feature space. Instead, we also maximise a diversity factor defined by maximising the maximum "information content" that each training sample can get from any sample in the clean validation set. In Table 7, we show an ablation study about the importance of this factor by redefining $\mathsf{Info}(.)$ in Eq. (7) with Eq. (6) (**Weight in Eq. (6)**). We also study the role of $\mathsf{Clean}(.)$ in Eq. (7) by optimising only the lower part of Eq. (7) ($\mathsf{Info}(.)$ **Only**). This ablation study is conducted on CIFAR10 and CIFAR100 under 0.4 asymmetric noise and 0.2 symmetric noise with imbalanced data. Overall, each component improves the performance compared to just naively optimising the weight in Eq. (6). Adding $\mathsf{Info}(.)$ and $\mathsf{Clean}(.)$ improves the model significantly. Naively selecting samples based on Eq. (6) facilitates the overfiting of the noisy-label samples, leading to confirmation bias. To mitigate this problem, $\mathsf{Clean}(.)$ limits the noise in the clean validation set, while $\mathsf{Info}(.)$ prevents the gradient to go toward a single wrong direction.

We also compare the validation set built with Eq. (7) with sets built with random sampling and most confident sampling based on the highest confidence score. Fig. 3 shows that most confident sampling shows inferior results compared to random sampling, but our method to build the validation set shows the best results.

Table 7: Test accuracy (%) on CIFAR10 and CIFAR100 under asymmetric and imbalanced noisy-label problems. The 1^{st} row shows the results of the optimisation of the weight (col. **Weight in Eq. (6)**) instead of Eq. (7). The 2^{nd} row shows the results of optimising the lower part of Eq. (7) (col. Info(.) **Only**) without the upper part of Eq. (7) Clean(.). The last row (**Whole Eq. (7)**) shows our final model result.

Weight in Eq. (6)	Info(.) **Only**	**Whole Eq. (7)**	**0.4 Asymmetric** Cifar10 WRN	Cifar10 RN29	**0.2 Uniform** Cifar10 RN32	Cifar10 RN32	Cifar10 RN32
Imb. Ratio			1	1	10	50	200
✓			91.0	56.6	68.8	37.6	23.4
	✓		92.1	89.3	89.0	79.1	65.9
		✓	94.1	89.8	90.1	80.1	66.6

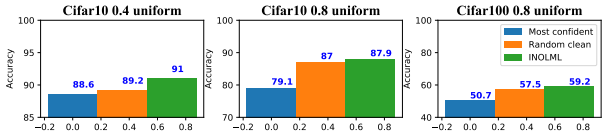


Figure 3: Accuracy (%) of our INOLML using different sample selection methods.

Traditional meta-learning approaches [13, 52] always keep the clean validation set separate from the training set, while our method iteratively extracts $D^{(v)}$ from the training set. It can be argued that this non-separation of the training and validation sets can cause confirmation bias to happen during training. Hence, we tested our approach in a scenario where the candidate samples to form the validation set is always separate from the training set during training. However, results showed that such separate validation set causes a 2% drop in accuracy, on average. This can be explained by the smaller size of the training set and the restriction in potential choices for validation samples.

A final discussion point is the time needed to run our approach. The Distill model takes around 5 and 29 hours to train the Resnet29 and WideResnet28-10 models, respectively. When integrating our method with Distill, training takes around 5.5 hours on Resnet29 and 31 hours on WideResnet28-10. Hence, our algorithm adds an 10% traning time overhead. Experiments are conducted on a single NVIDIA V100 GPU.

6 Conclusion

We presented a novel meta-learning approach, called INOLML, that automatically and progressively selects a pseudo clean validation set from a noisily-labelled training set. This selection is based on our proposed validation set utility that takes into account sample informativeness, class distribution balance, and label correctness. Our proposed method is more effective and practical than prior meta-learning approaches since we do not require manually-labelled samples to include in the validation set. Compared with other meta-learning approaches that do not require a manually labelled validation set, e.g. FSR or FaMUS, our method is demonstrated to be more robust to high noise rate problems and to achieve state-of-the-art results on several synthetic and realistic label noise benchmarks.

A limitation of our approach is that the model can suffer from confirmation bias as it is based on a single model. As future work, we will tackle this problem by incorporating co-training in our meta-learning algorithm. Another limitation is the greedy and complex bi-level optimisation to form the validation set in Eq. (7), which can be improved in two ways: 1) the complexity can be reduced by replacing the bi-level optimisation by a single-level optimisation, and 2) the greedy strategy can be replaced by a relaxation method to solve the combinatorial optimisation problem. Additionally, optimising the clean validation set once per epoch is not ideal since the validation set can be outdated by the end of epoch. This issue will be addressed by updating the clean validation set more regularly.

References

[1] G. Algan and I. Ulusoy. "Meta Soft Label Generation for Noisy Labels". In: *International Conference on Pattern Recognition*. 2021, pp. 7142–7148.

[2] G. Algan, ilkay Ulusoy, Şaban Gönül, Banu Turgut, and B. Bakbak. "Deep Learning from Small Amount of Medical Data with Noisy Labels: A Meta-Learning Approach". In: *International Conference on Retinopathy of Prematurity Ophthalmologic Approach*. 2021.

[3] Abhijit Bendale and T. Boult. “Towards Open Set Deep Networks”. In: *Conference on Computer Vision and Pattern Recognition*. 2016, pp. 1563–1572.

[4] Kaidi Cao, Yining Chen, Junwei Lu, Nikos Aréchiga, Adrien Gaidon, and Tengyu Ma. “Heteroskedastic and Imbalanced Deep Learning with Adaptive Regularization”. In: *International Conference on Learning Representations*. 2021.

[5] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Aréchiga, and Tengyu Ma. “Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss”. In: *Advances in Neural Information Processing Systems*. 2019.

[6] Can Chen, Shuhao Zheng, Xi Chen, Erqun Dong, Xue Liu, Hao Liu, and Dejing Dou. “Generalized Data Weighting via Class-level Gradient Manipulation”. In: *Advances in Neural Information Processing Systems*. 2021.

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin P. Murphy, and Alan Loddon Yuille. “Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs”. In: *IEEE Transaction on Pattern Analysis and Machine Intelligence* 40 (2018).

[8] Pengfei Chen, B. Liao, Guangyong Chen, and Shengyu Zhang. “Understanding and Utilizing Deep Neural Networks Trained with Noisy Labels”. In: *International Conference on Machine Learning*. 2019.

[9] Hsin-Ping Chou, Shih-Chieh Chang, Jia-Yu Pan, Wei Wei, and Da-Cheng Juan. “Remix: Rebalanced Mixup”. In: *European Conference on Computer Vision*. 2020.

[10] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. “Feature Space Augmentation for Long-Tailed Data”. In: *European Conference on Computer Vision*. 2020.

[11] Filipe R. Cordeiro, Vasileios Belagiannis, Ian D. Reid, and G. Carneiro. “PropMix: Hard Sample Filtering and Proportional MixUp for Learning with Noisy Labels”. In: *British Machine and Vision Conference*. 2021.

[12] M. Dehghani, Aliaksei Severyn, Sascha Rothe, and J. Kamps. “Avoiding Your Teacher’s Mistakes: Training Neural Networks with Controlled Weak Supervision”. In: *ArXiv* abs/1711.00313 (2017).

[13] M. Dehghani, Aliaksei Severyn, Sascha Rothe, and J. Kamps. “Learning to Learn from Weak Supervision by Full Supervision”. In: *ArXiv* abs/1711.11383 (2017).

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “ImageNet: A large-scale hierarchical image database”. In: *Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255.

[15] Charles Peter Elkan. “The Foundations of Cost-Sensitive Learning”. In: *International Joint Conference on Artificial Intelligence*. 2001.

[16] Erik Englesson and Hossein Azizpour. “Generalized Jensen-Shannon Divergence Loss for Learning with Noisy Labels”. In: *Advances in Neural Information Processing Systems*. 2021.

[17] Haojie Guo and Song Wang. “Long-Tailed Multi-Label Visual Recognition by Collaborative Training on Uniform and Re-balanced Samplings”. In: *Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15084–15093.

[18] Bo Han, Gang Niu, Jiangchao Yao, Xingrui Yu, Miao Xu, Ivor Tsang, and Masashi Sugiyama. “Pumpout: A meta approach for robustly training deep neural networks with noisy labels”. In: *ICML Workshop on Uncertainty and Robustness in Deep Learning*. 2019.

[19] 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*. 2018, pp. 8527–8537.

[20] Timothy M Hospedales, Antreas Antoniou, Paul Micaelli, and Amos J. Storkey. “Meta-Learning in Neural Networks: A Survey”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[21] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. “Learning Deep Representation for Imbalanced Classification”. In: *Conference on Computer Vision and Pattern Recognition*. 2016, pp. 5375–5384.

[22] Lee Jaehwan, Yoo Donggeun, and Kim Hyo-Eun. “Photometric Transformer Networks and Label Adjustment for Breast Density Prediction”. In: *International Conference on Computer Vision Workshops*. 2019.

[23] Lu Jiang, Di Huang, Mason Liu, and Weilong Yang. “Beyond synthetic noise: Deep learning on controlled noisy labels”. In: *International Conference on Machine Learning*. 2020.

[24] 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*. 2018.

[25] Bingyi Kang, Yu Li, Sai Nan Xie, Zehuan Yuan, and Jiashi Feng. “Exploring Balanced Feature Spaces for Representation Learning”. In: *International Conference on Learning Representations*. 2021.

[26] 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.

[27] Shyamgopal Karthik, Jérôme Revaud, and Chidlovskii Boris. “Learning From Long-Tailed Data With Noisy Labels”. In: *ArXiv* abs/2108.11096 (2021).

[28] Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. “NLNL: Negative Learning for Noisy Labels”. In: *International Conference on Computer Vision*. Oct. 2019, pp. 101–110.

[29] A. Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. 2009.

[30] Solomon Kullback and Richard A Leibler. “On information and sufficiency”. In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86.

[31] Kimin Lee, Sukmin Yun, Kibok Lee, Honglak Lee, Bo Li, and Jinwoo Shin. “Robust inference via generative classifiers for handling noisy labels”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3763–3772.

[32] Youngwan Lee and Jongyoul Park. “CenterMask: Real-Time Anchor-Free Instance Segmentation”. In: *Conference on Computer Vision and Pattern Recognition*. 2020, pp. 13903–13912.

[33] Junnan Li, Richard Socher, and Steven CH Hoi. “Dividemix: Learning with noisy labels as semi-supervised learning”. In: *International Conference on Learning Representations*. 2020.

[34] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. “Learning to learn from noisy labeled data”. In: *Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5051–5059.

[35] Wen Li, Limin Wang, Wei Li, E. Agustsson, and L. Gool. “WebVision Database: Visual Learning and Understanding from Web Data”. In: *ArXiv* abs/1708.02862 (2017).

[36] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. “Focal Loss for Dense Object Detection”. In: *International Conference on Computer Vision*. 2017, pp. 2999–3007.

[37] Jialun Liu, Yifan Sun, Chuchu Han, Zhaopeng Dou, and Wenhui Li. “Deep Representation Learning on Long-Tailed Data: A Learnable Embedding Augmentation Perspective”. In: *Conference on Computer Vision and Pattern Recognition*. 2020, pp. 2967–2976.

[38] Sheng Liu, Jonathan Niles-Weed, Narges Razavian, and Carlos Fernandez-Granda. “Early-learning regularization prevents memorization of noisy labels”. In: *Advances on Neural Information Processing Systems*. Vol. 33. 2020, pp. 20331–20342.

[39] I. Loshchilov and F. Hutter. “SGDR: Stochastic Gradient Descent with Warm Restarts”. In: *arXiv: Learning* (2017).

[40] Xingjun Ma, Yisen Wang, Michael E Houle, Shuo Zhou, Sarah Erfani, Shutao Xia, Sudanthi Wijewickrema, and James Bailey. "Dimensionality-Driven Learning with Noisy Labels". In: *International Conference on Machine Learning*. PMLR. 2018, pp. 3355–3364.

[41] Eran Malach and Shai Shalev-Shwartz. "Decoupling" when to update" from" how to update"". In: *Advances in Neural Information Processing Systems*. 2017, pp. 960–970.

[42] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. "Long-tail learning via logit adjustment". In: *International Conference on Learning Representations*. 2021.

[43] Qiguang Miao, Ying Cao, Ge Xia, Maoguo Gong, Jiachen Liu, and Jianfeng Song. "RBoost: Label noise-robust boosting algorithm based on a nonconvex loss function and the numerically stable base learners". In: *IEEE Transactions on Neural Networks and Learning Systems* 27.11 (2015), pp. 2216–2228.

[44] Baharan Mirzasoleiman, Kaidi Cao, and Jure Leskovec. "Coresets for Robust Training of Deep Neural Networks against Noisy Labels". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 11465–11477.

[45] Tam Nguyen, C Mummadi, T Ngo, L Beggel, and Thomas Brox. "SELF: learning to filter noisy labels with self-ensembling". In: *International Conference on Learning Representations*. 2020.

[46] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. "Towards Robust Learning with Different Label Noise Distributions". In: *arXiv preprint arXiv:1912.08741* (2019).

[47] Diego Ortego, Eric Arazo, Paul Albert, Noel E O'Connor, and Kevin McGuinness. "Multi-objective interpolation training for robustness to label noise". In: *Conference on Computer Vision and Pattern Recognition*. 2021, pp. 6606–6615.

[48] 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: *Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2233–2241.

[49] Foster J. Provost. *Machine Learning from Imbalanced Data Sets 101*. 2008.

[50] Mengye Ren, Wenyuan Zeng, Binh Yang, and R. Urtasun. "Learning to Reweight Examples for Robust Deep Learning". In: *International Conference on Machine Learning*. 2018.

[51] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2015), pp. 1137–1149.

[52] 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*. 2019.

[53] Haoliang Sun, Chenhui Guo, Qi Wei, Zhongyi Han, and Yilong Yin. "Learning to Rectify for Robust Learning with Noisy Labels". In: *Pattern Recognition* (2021), p. 108467.

[54] Yanmin Sun, Mohamed S. Kamel, Andrew Wong, and Yang Wang. "Cost-sensitive boosting for classification of imbalanced data". In: *Pattern Recognition* 40 (Dec. 2007), pp. 3358–3378.

[55] Antti Tarvainen and Harri Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results". In: *Advances in Neural Information Processing Systems*. 2017, pp. 1195–1204.

[56] Nidhi Vyas, Shreya Saxena, and T. Voice. "Learning Soft Labels via Meta Learning". In: *arXiv:2009.09496* (2020).

[57] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Jun Hao Liew, Sheng Tang, Steven C. H. Hoi, and Jiashi Feng. "The Devil is in Classification: A Simple Framework for Long-tail Instance Segmenta-

tion". In: *European Conference on Computer Vision*. 2020.

[58] Xinshao Wang, Yang Hua, Elyor Kodirov, and N. Robertson. "IMAE for Noise-Robust Learning: Mean Absolute Error Does Not Treat Examples Equally and Gradient Magnitude's Variance Matters". In: *arXiv: Learning* (2019).

[59] Xinshao Wang, Yang Hua, Elyor Kodirov, and Neil M Robertson. "IMAE for noise-robust learning: Mean absolute error does not treat examples equally and gradient magnitude's variance matters". In: *arXiv preprint arXiv:1903.12141* (2019).

[60] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. "Learning to Model the Tail". In: *Advances on Neural Information Processing Systems*. 2017.

[61] Yisen Wang, Weiyang Liu, Xingjun Ma, James Bailey, Hongyuan Zha, Le Song, and Shu-Tao Xia. "Iterative learning with open-set noisy labels". In: *Conference on Computer Vision and Pattern Recognition*. 2018, pp. 8688–8696.

[62] Yisen Wang, Xingjun Ma, Zaiyi Chen, Yuan Luo, Jinfeng Yi, and James Bailey. "Symmetric cross entropy for robust learning with noisy labels". In: *International Conference on Computer Vision*. 2019, pp. 322–330.

[63] Tong Wei, Jiang-Xin Shi, Wei-Wei Tu, and Yu-Feng Li. "Robust Long-Tailed Learning under Label Noise". In: *ArXiv* abs/2108.11569 (2021).

[64] Tong Wu, Ziwei Liu, Qingqiu Huang, Yu Wang, and Dahua Lin. "Adversarial Robustness under Long-Tailed Distribution". In: *Conference on Computer Vision and Pattern Recognition*. 2021, pp. 8655–8664.

[65] Tong Xiao, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang. "Learning from massive noisy labeled data for image classification". In: *Conference on Computer Vision and Pattern Recognition*. 2015, pp. 2691–2699.

[66] Youjiang Xu, Linchao Zhu, Lu Jiang, and Yi Yang. "Faster Meta Update Strategy for Noise-Robust Deep Learning". In: *Conference on Computer Vision and Pattern Recognition*. 2021.

[67] Kun Yi and Jianxin Wu. "Probabilistic End-To-End Noise Correction for Learning With Noisy Labels". In: *Conference on Computer Vision and Pattern Recognition*. June 2019, pp. 7010–7018.

[68] 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*. 2019.

[69] Xiyu Yu, Tongliang Liu, Mingming Gong, and Dacheng Tao. "Learning with biased complementary labels". In: *European Conference on Computer Vision*. 2018, pp. 68–83.

[70] Bodi Yuan, Jianyu Chen, Weidong Zhang, Hung-Shuo Tai, and Sara McMains. "Iterative cross learning on noisy labels". In: *IEEE Winter Conference on Applications of Computer Vision*. 2018, pp. 757–765.

[71] Yuhang Zang, Chen Huang, and Chen Change Loy. "FASA: Feature Augmentation and Sampling Adaptation for Long-Tailed Instance Segmentation". In: *International Conference on Computer Vision*. 2021.

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

[73] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. "Is Faster R-CNN Doing Well for Pedestrian Detection?" In: *European Conference on Computer Vision*. 2016.

[74] Weihe Zhang, Yali Wang, and Yu Qiao. "MetaCleaner: Learning to hallucinate clean representations for noisy-labeled visual recognition". In: *Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7373–7382.

[75] Xiao Zhang, Zhiyuan Fang, Yandong Wen, Zhifeng Li, and Yu Qiao. "Range Loss for Deep Face Recognition with Long-Tailed Training Data". In: *International Conference on Computer Vision*. 2017, pp. 5419–5428.

[76] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. “Deep Long-Tailed Learning: A Survey”. In: *arXiv:2110.04596* (2021).

[77] Yikai Zhang, Songzhu Zheng, Pengxiang Wu, Mayank Goswami, and Chao Chen. “Learning with Feature Dependent Label Noise: a Progressive Approach”. In: *International Conference on Feature Representations* (2021).

[78] Zhilu Zhang and Mert Sabuncu. “Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels”. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018.

[79] Zizhao Zhang and Tomas Pfister. “Learning Fast Sample Re-weighting Without Reward Data”. In: *International Conference on Computer Vision*. 2021.

[80] Zizhao Zhang, Han Zhang, Sercan Ö. Arik, Honglak Lee, and Tomas Pfister. “Distilling Effective Supervision From Severe Label Noise”. In: *Conference on Computer Vision and Pattern Recognition*. 2020, pp. 9291–9300.

[81] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. “BBN: Bilateral-Branch Network With Cumulative Learning for Long-Tailed Visual Recognition”. In: *Conference on Computer Vision and Pattern Recognition* (2020), pp. 9716–9725.

[82] Zhi-Hua Zhou and Xu-Ying Liu. “Training cost-sensitive neural networks with methods addressing the class imbalance problem”. In: *IEEE Transactions on Knowledge and Data Engineering* 18.1 (2006), pp. 63–77.

A Algorithm of the proposed method

Algorithm 1 Training procedure of the proposed INOLML.

procedure TRAINING($\mathcal{D}$, η, T, $\widetilde{T}$, $T^{(u)}$, $\widetilde{\eta}$, κ, N, M, K, C)
  ▷ $\mathcal{D}$: *noisy dataset* ◁
  ▷ η: *learning rate per iteration* ◁
  ▷ T: *total number of iterations* ◁
  ▷ $\widetilde{T}$: *minimum number of iterations before updating the moving average robust labels* ◁
  ▷ $T^{(u)}$: *interval between updates* ◁
  ▷ $\widetilde{\eta}$: *learning rate threshold* ◁
  ▷ κ, N, M, K, C: *hyper-parameters* ◁

  Warmup model $f_\theta(.)$ with $\ell_{\mathrm{CE}}(.)$ from $\mathcal{D}$
  $\mathcal{D}^{(c)} = \mathsf{PseudoCleanDetector}\,(\mathcal{D})$ using Equation 3
  Initialise moving average robust label $\{\tilde{\mathbf{y}}_i\}_{i=1}^{|\mathcal{D}^{(c)}|}$ of samples in $\mathcal{D}^{(c)}$
  Initialise $\mathcal{D}^{(v)}$ and $\mathcal{D}^{(t)}$ from $\mathcal{D}^{(c)}$ using Equation 7
  Reinitialize model $f_\theta(.)$
  for $t = 1$ to T **do**
    Meta-learn to train θ, ω and λ using Equation 1
    Update $\{\tilde{\mathbf{y}}_i\}_{i=1}^{|\mathcal{D}^{(c)}|}$ of samples in $\mathcal{D}^{(c)}$ if $t > \widetilde{T}$ and $\eta_t < \widetilde{\eta}$
    Update $\mathcal{D}^{(c)} = \mathsf{PseudoCleanDetector}\,(\mathcal{D})$ using Equation 3 if $\eta_t = 0$
    if $T^{(u)} \equiv t \mod T^{(u)}$ **then**
      Update $\mathcal{D}^{(v)}$ and $\mathcal{D}^{(t)}$ from $\mathcal{D}^{(c)}$ using Equation 7
  return the trained model parameter θ

B Implementation Details

All CIFAR experiments use batches of size 100, which are trained on a single GPU. Similar to the Distill noise model [80], we use $p = 5, k = 20$ for CIFAR experiments, except the ones with the imbalance setting.

For Red Mini-ImageNet experiments, we trained the model on a single GPU with batches of size 100, with $p = 5, k = 10$.

For the WebVision experiment, we use $p = 4, k = 8$ with 4 NVIDIA V100 GPU and batches of size 16. All experiments use $N = 200, K = 50, \kappa = 0.9$.

C Additional Results of Symmetric Noise on CIFAR Datasets

We provide additional symmetric noise results of our proposed method and the Distill model [80] in Table 8. Note that our method is markedly better than Distill, particularly for the simpler model (RN29) with few samples per class (1 and 5) in the validation set. For the more complex model (WRN) and large validation set (10 samples per class), our method is still better than Distill, except for CIFAR100 at 0.8 symmetric noise rate.

Table 8: Test accuracy (in %) comparison between our method ('INOLML') and the Distill noise ('DN') on symmetric noise using 1, 5 and 10 samples per class in the validation set on two backbone models: Resnet29 ('RN29') and Wideresnet28-10 ('WRN'). The results of the Distill model with WideResnet28-10 are collected from [80]. Recall that the Distill needs a clean set, while INOLML works with a pseudo-clean set.

Method	**Val. Set size**	**Dataset**					
		CIFAR10			**CIFAR100**		
		0.2	**0.4**	**0.8**	**0.2**	**0.4**	**0.8**
DN-RN29	1	87.0 ± 0.5	85.3 ± 0.5	FAIL	58.9 ± 0.5	55.8 ± 0.7	FAIL
INOLML-RN29		90.3 ± 0.2	89.1 ± 0.5	79.1 ± 0.3	65.9 ± 0.2	61.5 ± 0.2	55.1 ± 0.1
DN-RN29	5	90.7 ± 0.3	89.0 ± 0.3	83.5 ± 0.2	62.6 ± 0.4	58.8 ± 0.5	48.5 ± 0.5
INOLML-RN29		90.9 ± 0.2	90.9 ± 0.1	87.4 ± 0.2	66.6 ± 0.1	65.7 ± 0.1	59.0 ± 0.5
DN-RN29	10	91.0 ± 0.2	89.2 ± 0.1	87.0 ± 0.1	63.7 ± 0.2	60.5 ± 0.2	57.5 ± 0.5
INOLML-RN29		92.2 ± 0.1	91.0 ± 0.1	87.9 ± 0.2	67.1 ± 0.1	66.3 ± 0.1	59.2 ± 0.2
DN-WRN	1	95.4 ± 0.6	94.5 ± 1.0	87.9 ± 5.1	77.4 ± 0.4	75.5 ± 1.1	62.1 ± 1.2
INOLML-WRN		96.0 ± 0.2	95.9 ± 0.2	94.3 ± 0.2	81.6 ± 0.2	79.5 ± 0.2	73.6 ± 0.3
DN-WRN	5	96.4 ± 0.0	95.5 ± 0.6	91.8 ± 3.0	80.4 ± 0.5	79.6 ± 0.3	73.6 ± 1.5
INOLML-WRN		96.4 ± 0.1	96.2 ± 0.1	94.6 ± 0.2	82.2 ± 0.2	81.5 ± 0.2	74.5 ± 0.2
DN-WRN	10	96.2 ± 0.2	95.9 ± 0.2	93.7 ± 0.5	81.2 ± 0.7	80.2 ± 0.3	**75.5 ± 0.2**
INOLML-WRN		**96.9 ± 0.1**	**96.6 ± 0.1**	**95.0 ± 0.2**	**82.0 ± 0.2**	**81.3 ± 0.2**	74.7 ± 0.1