Abstract

Many state-of-the-art noisy-label learning methods rely on learning mechanisms that estimate the samples’ clean labels during training and discard their original noisy labels. However, this approach prevents the learning of the relationship between images, noisy labels and clean labels, which has been shown to be useful when dealing with instance-dependent label noise problems. Furthermore, methods that do aim to learn this relationship require cleanly annotated subsets of data, as well as distillation or multi-faceted models for training. In this paper, we propose a new training algorithm that relies on a simple model to learn the relationship between clean and noisy labels without the need for a cleanly labelled subset of data. Our algorithm follows a 3-stage process, namely: 1) self-supervised pre-training followed by an early-stopping training of the classifier to confidently predict clean labels for a subset of the training set; 2) use the clean set from stage (1) to bootstrap the relationship between images, noisy labels and clean labels, which we exploit for effective relabelling of the remaining training set using semi-supervised learning; and 3) supervised training of the classifier with all relabelled samples from stage (2). By learning this relationship, we achieve state-of-the-art performance in asymmetric and instance-dependent label noise problems.

1 Supported by Australian Research Council through grants DP180103232 and FT190100525.

1. Introduction

Supervised deep learning has had great success generating effective classification models from sets of labelled training data [23, 25]. Modern deep learning models require large-scale datasets to achieve state-of-the-art (SOTA) results [37, 38]. However, real-world large-scale datasets, such as those collected from search engines or available from hospitals and clinics, tend to have a non-negligible amount of instance-dependent label noise (IDN) [51]. Existing methods often attempt to address instance-independent label noise (IIN), such as symmetric or asymmetric noise [14, 55, 63]. Handling the IDN present in large-scale real-world datasets has become one of the main research problems in the field.

When naively trained with noisy-labelled data, deep learning models generalise poorly because they can easily overfit the incorrectly labelled samples [60]. Many methods have been developed for handling label noise, with SOTA approaches relying on sample relabelling mechanisms. These strategies are based on techniques to estimate the relationship between images and clean labels, and after relabelling, the old noisy labels are discarded [28, 46, 63]. However, to model how different image features and noisy labels affect the mislabelling process in IDN, we need to estimate the relationship between images, clean labels and noisy labels [16, 63]. Some methods have attempted to model this relationship with noise-transition matrices and corrective layers for asymmetric noise [14, 50, 53] or part-dependant noise in place of instance-dependant noise [54], but they failed to achieve SOTA results.

Rather than the usual noisy-label learning setting, where a set of cleanly annotated samples is not available, some methods assume the existence of a subset of training data containing images, clean labels and noisy labels [16, 20, 50]. By training a model that predicts clean labels from both images and noisy labels (see right of Figure 1), these methods are able to learn the relationship between image features, noisy labels and clean labels, allowing them to model IDN and more effectively relabel noisy samples. However, it can be expensive, difficult and time-consuming to obtain a clean subset of data with noisy labels and clean labels that is representative of the instance-dependant noise in the dataset. Furthermore, these methods require distillation to a more standard model (such as the one on the left of Figure 1) for evaluation on samples without labels.
In this paper, we introduce a new algorithm to learn the relationship between images and their clean and noisy labels without using any clean-label set. Our algorithm follows a 3-stage process (see Fig. 2): 1) **Bootstrapping**: self-supervised pre-training followed by an early-stopping training of the classifier that receives images and ‘null’ labels as input and predicts the noisy labels as output – this stage forms a subset of predicted clean labels for the second stage of training; 2) **Semi-supervised Learning**: use this predicted clean subset to learn the relationship between images, noisy labels and clean labels, which we exploit for an effective, explicit relabelling of the remaining training set; and 3) **Final training**: supervised training of the classifier using the relabelled samples. The main contributions of this paper are:

- An effective three-stage training algorithm designed to address instance-dependent label noise by learning the relationship between images and their clean and noisy labels – using a noise-transition sample balancing scheme, explicitly relabelling training samples and without requiring a cleanly annotated training set;

- A method that reaches SOTA asymmetric and instance dependent label noise results using a simple single-model architecture, unlike DivideMix [28] (and its derivatives such as [10, 21, 24, 41, 66]) that require a more complex 2-model architecture.

- A ‘label dropping’ strategy that removes the need for distillation to a standard model and allows predictions to be made on samples with and without noisy labels;

2. Related Work

2.1. Noisy Label Learning Based on Semi-supervised Learning (SSL)

Many SOTA methods in noisy label learning use SSL techniques to perform label correction and consistency regularization. DivideMix [28] and variants [10, 41, 66] perform sample re-labelling with co-teaching and MixMatch data augmentation [6]. ELR+ [32] and PES [5] similarly use MixUp [61] based SSL on top of a regularising loss functions. These techniques are effective for symmetric and asymmetric noise, but are dependent on carefully tuned hyperparameters and cautious integration of the sample relabelling and SSL techniques used.

To identify the incorrectly labelled samples for forming the unlabelled set, many methods depend on loss-separation techniques, relying on the ability of deep networks to learn clean samples faster than noisy samples [2], which leads to lower loss values for clean samples after a few stages of training [1, 7, 17, 28, 57]. FINE alternatively uses eigen-decomposition to separate samples in feature space [21]. However, such automatic and dynamic identification of noisy-label samples is a brittle process that tends to fail in challenging noisy-label learning scenarios, such as instance-dependent label noise, because the differences between hard clean-label and noisy-label samples can be subtle during early training stages.

2.2. Label Transition Estimation Methods

Many methods attempt to model the class-dependent asymmetric noise, such as with a label transition matrix [55], by learning noise-adaption layers and performing loss-correction [14, 36], or by using reconstruction error as a consistency objective [39]. However, these methods are not as competitive as SSL approaches in Sec. 2.1 because they generally do not address instance-dependent noise and have limited mechanisms to make use of mislabeled samples. Methods that attempt to handle semantic noise by estimating instance-based transition matrices can in principle deal with semantic noise [14] and can provide guarantees on convergence and bounds on generalization error [63], but they do not provide SOTA results in practice.

2.3. Methods Based on Clean Validation Sets

Alternatively, researchers have explored learning methods that require the existence of a small, additional clean set of data to learn from. For instance, many meta-learning strategies require clean validation samples to adjust the weights of each training sample [40], to simulate regular training with synthetic noise labels [29], or to learn an explicit weighting function [43], or to estimate the noise transition matrix [52]. Other noisy label learning algorithms rely on clean sets of data for which noisy labels
and clean labels exist for samples, so that the relationship between image features, noisy labels and clean labels can be learned [16, 20, 50], using fully-connected neural networks to predict true labels from images and their noisy labels. Together, these methods show the utility of representative clean sets of data to the noisy label problem, but they rely on manual labelling, which can be expensive and time-consuming to collect, particularly in domains that require a high degree of expertise for labellers, such as medical imaging [33, 65].

Other methods aim to dynamically construct a pseudo-clean set out of high-confidence samples, such as by using K-Nearest Neighbours to identify related samples in the feature space [4, 35], or by using meta-learning to identify a dictionary of dynamically updating valuable training samples [64].

2.4. Background Material

Our method relies on many techniques previously developed in the field, such as self-supervision, SSL and data augmentation. Recently, self-supervised methods such as SimCLR [8, 9] and SCAN [49] have been used for pre-training, or as auxiliary objectives in noisy-label learning tasks [41, 66], due to their ability to learn high-level features from noisy data without the risk of overfitting incorrect labels. In this paper, we utilize the SSL method FixMatch [45], which uses pseudo-labelling thresholds and consistency between strong and weak augmentations to regularize training through consistency regularization [3, 24, 48] and entropy minimization [15, 26]. Strong data augmentation strategies, such as RandAugment [12], AutoAugment [11] and MixUp [61] have been shown to be effective for regularising training, preventing overfitting and dramatically improving the tolerance of algorithms to higher noise levels [1, 28, 34].

3. Methodology

Our algorithm (see Fig. 2) is motivated by the objective to train a model that can accurately relabel samples by predicting true labels from images and noisy labels without requiring clean-labelled data. The stages of the proposed algorithm are: 1) Bootstrapping: perform self-supervised pre-training and early-stopping training to identify a representative, clean subset of samples, 2) SSL: learn the instance-dependent noise relationship from the clean set (from stage 1) and use it to relabel the remaining noisy samples, and 3) Final Training: use the relabelled samples from stage 2 to train a final, regularized classifier.

For the methods described below, assume the availability of a training set $D = \{(x_i, \tilde{y}_i)\}_{i=1}^{|D|}$, where $x \in \mathcal{X} \subset \mathbb{R}^{H \times W \times R}$ denotes an image of size $H \times W$ with $R$ colour channels, and $\tilde{y} \in \mathcal{Y} \subset \{0, 1\}^{|Y|}$ represents the one-hot noisy label. Our model, referred to as ‘modified’, receives an image and the noisy label at the input and outputs a clean label classification distribution, with $f_\theta : \mathcal{X} \rightarrow \Delta_{|Y| - 1}$, where $\Delta_{|Y| - 1}$ represents the $|Y| - 1$ probability simplex, and $\theta \in \Theta$ denotes the model parameters. Note that we also consider a ‘normal’ model, which is a model that takes an image input and outputs a classification, with $f_\theta : \mathcal{X} \rightarrow \Delta_{|Y| - 1}$.

3.1. Bootstrapping

The goal of this first stage is to train a model that accepts images and noisy labels and predicts clean labels, however at the beginning of training we only have access to images and noisy labels from $D$. Following [49, 66], we start with SimCLR pre-training [8], which allows us to learn an initial feature representation from $D$ without the risk of overfitting.

Next, we take the pre-trained model above to learn a clas-
sifier with early-stopping and small learning rate with
\[
\theta^* = \arg\min_{\theta} \frac{1}{|D|} \sum_{(x_i, y_i) \in D} E_{a(.) \sim A_S} \left[ \ell_{CE}(\hat{y}_i, f_\theta(a(x_i), 0_{|Y|})) \right],
\]
where \(a(.)\) is a strong data augmentation sampled from the set of strong data augmentation functions \(A_S\), \(\ell_{CE}(.)\) denotes the cross-entropy loss function, and \(0_{|Y|}\) is a ‘null’ label vector with \(|Y|\) zeros.

Then, we use our trained model to generate a prediction distribution for all training samples using test-time weak augmentation, as follows:
\[
\hat{y}_i = E_{a(.) \sim A_W} \left[ f_{\theta'}(a(x_i), 0_{|Y|}) \right],
\]
where \(a(.)\) is a weak augmentation sampled from the set of weak data augmentation functions \(A_W\). We also have dropout enabled during this evaluation process. By using dropout and multiple weak augmentations to evaluate samples, we penalise samples with highly confident but inconsistent predictions [63]. The confidence prediction for \(x_i\) is given by \(\max_{c \in Y} \hat{y}_i(c)\).

We then split the dataset into a confident clean set and a noisy set. However, if we naively select the most confident samples, we will disproportionately select samples from easy classes, and samples whose original noisy labels were correct. We want the clean set to contain representative samples from all classes and noise transitions in the dataset, as the upcoming SSL process can only learn noise transitions which are present in the clean set. To achieve this, we propose noise-transition sample balancing that first estimates the dataset’s noise transition matrix \(T\) by using the noisy labels and predicted labels for the 90% of most confident predictions per class, where \(T_{ij}\) represents the estimated proportion of samples in the dataset with the noisy label \(i\) and clean label \(j\). We then select the \(K \times |Y| \times T_{ij}\) most confident samples from each noise transition, as well as any other samples \(x_i\) where \(\max_{c \in Y} \hat{y}_i(c) > \tau\), where \(K\) is a hyperparameter controlling the minimum fraction of samples from each subset we want to select, and \(\tau\) is a hyperparameter controlling how confident a prediction needs to be before it is guaranteed to be selected.

We note that this process does not guarantee that all instance-dependent relationships between image features and noisy label transitions are captured, but in practice this ensures a large coverage of the different noise transitions present in the dataset. This ‘noise-based’ balancing approach can be contrasted against the usual class-based balancing typically seen in noisy-label learning methods, where samples are selected to balance the number of samples per class. The initial clean set will contain the samples with both the noisy and estimated clean labels with \(C = \{(x_i, \tilde{y}_i, \hat{y}_i)| (x_i, \hat{y}_i) \in D\}\), and the initial noisy set will contain the samples and noisy labels as \(U = \{(x_i, \tilde{y}_i)| (x_i, \tilde{y}_i, \hat{y}_i) \notin C\) and \((x_i, \tilde{y}_i) \in D\)\.

### 3.2. Semi-Supervised Learning (SSL) for Noisy Label Correction

The next stage of our framework takes the initial clean set \(C\) and the initial noisy set \(U\) to train the SSL model, where images and noisy labels (the model inputs) are present for all samples, and the true labels (the model output) are present for samples in \(C\). Our SSL algorithm is based on FixMatch (see Fig. 3), which achieves competitive performance by focusing on consistency regularization and entropy minimization [45]. We do not reinitialize the network before semi-supervised learning, instead using the bootstrapping process as a form of warmup.

![Figure 3. The noisy-label FixMatch algorithm.](image)

Because we are learning to predict the ‘true’ labels of samples from images and noisy labels, our model is able to learn a joint distribution between the three, similar to works that require a clean set [16, 20, 50]. However, if we were to train our model to always make predictions from images and noisy labels, our model would no longer be able to make meaningful predictions on samples without associated noisy labels, such as those in a ‘test’ set. To remedy this, we randomly ‘drop’ the one-hot noisy label from samples 50% of the time, replacing it with a ‘null’ label. By doing this, our model also learns a direct relationship between images and true labels without depending on the noisy labels, allowing us to evaluate test samples by passing them into the model alongside a ‘null’ label. In our implementation of FixMatch, we drop the noisy label of supervised samples and strong augmentations of unsupervised samples 50% of the time. However, we keep the noisy labels for the weakly augmented unsupervised samples as we want to always use these noisy labels to predict higher accuracy pseudo-labels, given that the loss is not backpropagated along these samples (see Fig. 3). We experiment with this decision in the
we re-label the whole training set to form what we refer to as ‘RoG’ noise. We also test our system with synthetic label noise. The first type of noise introduced by Lee et al. [27], where incorrect predictions are mislabelled at higher rates than samples far from decision boundaries. The second type is the semantic noise [62], where confusing samples near decision boundaries are mislabelled at higher rates than samples far from decision boundaries. In (4) that our model uses the learned relationship between images, noisy labels and ‘true’ labels to relabel the remaining noisy samples.

3.3. Final Model Training

After forming $\mathcal{D}$, we train a final model with strong augmentations and MixUp, due to their robustness to any noise that may remain in the re-labelled training set [11]. MixUp is applied on both images and their noisy labels together. After applying Mixup, we randomly replace the noisy label with a ‘null’ label in 50% of samples.

4. Experiments

4.1. Data sets

To investigate our method, we perform experiments using the CIFAR-10, CIFAR-100 [22], Animal10N [46], and Webvision [31] datasets. CIFAR-10 and CIFAR-100 consist of 50,000 training and 10,000 testing images of size $32 \times 32$ pixels, with 10 and 100 classes respectively. As CIFAR-10 and CIFAR-100 do not contain label noise, we follow the literature and perform experiments with different types of controlled, synthetic label noise. The first type of noise is the Polynomial Margin Diminishing (PMD) semantic noise [62], where confusing samples near decision boundaries are mislabelled at higher rates than samples far from decision boundaries. The second type is the semantic noise introduced by Lee et al. [27], where incorrect predictions from trained VGG [44], DenseNet [19], and ResNet [18] models are used to generate mislabelled samples (which we refer to as ‘RoG’ noise). We also test our system with symmetric noise rates of \{20\%, \ 50\%, \ 80\%, \ 90\%\} and asymmetric noise using the mapping from [28][36] with 40\% rate.

The Animal10N dataset [46] consists of 50,000 training images and 5,000 testing images of size $64 \times 64$, consisting of five pairs of semantically similar classes. Images are collected for each label using online search engines, resulting in incorrect classifications for an estimated 6\% to 10\% of samples. Finally, we test with Mini-Webvision, which consists of the 65,944 samples which make up the first 50 classes of the Webvision dataset that contains images collected from the internet. Images are resized to $256 \times 256$, and the corresponding 50 classes in the ILSVRC12 dataset [13] are also used for validation.

4.2. Implementation

Following contemporary work [10][28][32], we use a PreAct-ResNet18 (PRN18) network [18] as our backbone model for CIFAR10 and CIFAR100 experiments, use a VGG19 model [44] as our backbone for Animal10N, and use an Inception-ResnetV2 model [47] as our backbone for Webvision. For weak augmentations, we use horizontal flipping and random cropping, and for strong augmentations we used AutoAugment [11], followed by horizontal flipping and random cropping. For all experiments, we perform bootstrapping with strong augmentations and perform pseudo-labeling with 25 weak augmentations. Final model training is done using Mixup [61] (with $\alpha = 1$). For all stages of training, we use stochastic gradient descent, with additional information about the optimizer and training schedule hyperparameters provided in the supplementary material. For Webvision, we additionally use label smoothing (with $\epsilon = 0.1$). Following existing implementations of FixMatch (such as by TorchSSL [59]), we use Exponential Moving Average (EMA) models to perform temporal ensembling [24]. For fair comparison with existing noisy-label learning methods, we shorten the training schedules typically used by FixMatch implementations (we use 100,000 training iterations with $\mu = 4$ rather than 1,000,000 iterations with $\mu = 8$).

We report two sets of results for our experiments in order to understand how the use of noisy labels during semi-supervised learning can improve results: ‘Normal Model’ and ‘Modified Model’. In both cases, we utilise the training procedure outlined in Figure 2, however:

1) In ‘Normal Model’, we use a standard model which only accepts image inputs. Because this type of model does not aim to learn the relationship between noisy labels and clean labels, we perform class-based balancing rather than the proposed noise-based balancing.

2) In ‘Modified Model’, we use noise-based balancing, as well as the model that accepts images and noisy labels, as described in Sec. 3.

For all our experiments, we also report the accuracy obtained from test-time augmentation, where 25 weakly aug-
We next turn our attention to the synthetic ‘Polynomial Margin Diminishing (PMD)’ \cite{62} and ‘RoG’ \cite{27} instance-dependent noisy label benchmarks based on CIFAR10 and CIFAR100 in Tables 3 and 4. In all cases considered in these two tables, our accuracy results are substantially higher than by other approaches, even without our modified model. When we do use a modified model for these synthetic instance-dependent noisy-label benchmarks, we find mixed results, with performance often decreasing, potentially due to learned noise transitions not generalizing as well as they do in real-world instance-dependent noise datasets.

Finally, in Table 5 we show the results of our method on the synthetically constructed symmetric and asymmetric noise for CIFAR10 and CIFAR100. These noise types are rare in practice, but they are common noisy-label benchmarks so we include them here for completeness. We see that our method is competitive with the state-of-the-art on CIFAR10 symmetric and asymmetric noise, despite featuring fewer mechanisms designed to address these types of noise. We particularly note 40% asymmetric noise, which benefits from the modified model due to noisy labels greatly limiting the set of feasible samples for each image, allowing us to exceed the state-of-the-art. In contrast to this, we report our results on symmetric label noise in CIFAR100, where existing methods perform better than ours. In all cases, our modified model is able to take advantage of the noise to provide more accurate relabelling, but the regularisation strategies that other methods use work well under the assumption of symmetric noise and provide stronger results. Perfectly symmetric noise over 100 classes is rare in practice though, and our results show universally strong performance on real-world instance-dependent datasets.

### 4.4. Predictions with Noisy Labels

A unique feature of our method is that our final model can be used to predict the labels of samples with and without noisy labels. In some applications, e.g., tagged image classification, images at test-time may also have noisy labels associated with them, which our model can use to improve classification performance. To show this, we generate artificial noisy labels for all the samples in the CIFAR10 symmetric and asymmetric noise, despite featuring fewer mechanisms designed to address these types of noise. We particularly note 40% asymmetric noise, which benefits from the modified model due to noisy labels greatly limiting the set of feasible samples for each image, allowing us to exceed the state-of-the-art. In contrast to this, we report our results on symmetric label noise in CIFAR100, where existing methods perform better than ours. In all cases, our modified model is able to take advantage of the noise to provide more accurate relabelling, but the regularisation strategies that other methods use work well under the assumption of symmetric noise and provide stronger results. Perfectly symmetric noise over 100 classes is rare in practice though, and our results show universally strong performance on real-world instance-dependent datasets.
Table 3. Test accuracy (%) for Polynomial Margin Diminishing Noise [62]. Top methods are in \textbf{bold}.

<table>
<thead>
<tr>
<th>Data set</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Type</td>
<td>Type-I 35%</td>
<td>Type-II 35%</td>
</tr>
<tr>
<td>Cross-Entropy</td>
<td>78.11</td>
<td>76.65</td>
</tr>
<tr>
<td>PLC [63]</td>
<td>82.80</td>
<td>81.54</td>
</tr>
<tr>
<td>Ours (Normal Model)</td>
<td>94.06</td>
<td>93.25</td>
</tr>
<tr>
<td>+ Test-Time Aug.</td>
<td>94.72</td>
<td>93.79</td>
</tr>
<tr>
<td>Ours (Modified Model)</td>
<td>94.00</td>
<td>93.76</td>
</tr>
<tr>
<td>+ Test-Time Aug.</td>
<td>94.39</td>
<td>94.19</td>
</tr>
</tbody>
</table>

Table 4. Test accuracy (%) for the RoG label noise benchmark [27], where baseline results are from [27]. Top methods are in \textbf{bold}.

<table>
<thead>
<tr>
<th>Data set</th>
<th>CIFAR-10</th>
<th>CIFAR-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise Type</td>
<td>Sym.</td>
<td>Asym.</td>
</tr>
<tr>
<td>Method / Noise Ratio</td>
<td>20%</td>
<td>50%</td>
</tr>
<tr>
<td>Cross-Entropy [28]</td>
<td>82.7</td>
<td>57.9</td>
</tr>
<tr>
<td>ELR [32]</td>
<td>95.8</td>
<td>94.8</td>
</tr>
<tr>
<td>DivideMix [28]</td>
<td>95.7</td>
<td>94.4</td>
</tr>
<tr>
<td>AugDesc [34]</td>
<td>96.3</td>
<td>95.4</td>
</tr>
<tr>
<td>ContrastToDivide [66]</td>
<td>96.4</td>
<td>95.3</td>
</tr>
<tr>
<td>PropMix [10]</td>
<td>96.09</td>
<td>95.53</td>
</tr>
<tr>
<td>Ours (Normal Model)</td>
<td>93.26</td>
<td>92.05</td>
</tr>
<tr>
<td>+ Test-Time Aug.</td>
<td>93.87</td>
<td>92.66</td>
</tr>
<tr>
<td>Ours (Modified Model)</td>
<td>89.46</td>
<td>90.97</td>
</tr>
<tr>
<td>+ Test-Time Aug.</td>
<td>90.25</td>
<td>91.85</td>
</tr>
</tbody>
</table>

Table 5. Test accuracy (%) for all competing methods on CIFAR-10 and CIFAR-100 under symmetric and asymmetric noises. Results from related approaches are as presented in [28] and [53]. Top methods within 1% are in \textbf{bold}.

<table>
<thead>
<tr>
<th>Test Set</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Noisy Labels</td>
<td>95.85</td>
</tr>
<tr>
<td>With Noisy Labels</td>
<td>97.59</td>
</tr>
</tbody>
</table>

4.5. Clean Set Selection

On the left of Figure 4, we show the distribution of confidences of our model after bootstrapping, and see that the highest confidence predictions are almost entirely for samples whose predicted ‘true’ label (\(\hat{y}\)) is correct, which allows us to select highly accurate clean sets for SSL. On the right, we show the percentage of the selected samples for the clean set that were originally clean (i.e. samples for which \(\hat{y}\) matches the true label) if no form of noise balancing is performed. We see that if we did not perform noise balancing, the selected clean set would disproportionately consist of clean samples, which would cause a degenerate relationship between noisy labels and clean labels to be learned during the SSL stage of training, preventing accurate relabeling of samples from other noise transitions.
Table 7. Predictions made by our model with different noisy labels for a testing sample (showing a dog) in Asym 0.4 noise for CIFAR10. '-' represents using a null label in place of a noisy label.

<table>
<thead>
<tr>
<th>Noisy Label</th>
<th>Prediction (Confidence)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airplane</td>
<td>Bird (65.39%)</td>
</tr>
<tr>
<td>Automobile</td>
<td>Automobile (62.75%)</td>
</tr>
<tr>
<td>Bird</td>
<td>Bird (98.30%)</td>
</tr>
<tr>
<td>Cat</td>
<td>Dog (94.51%)</td>
</tr>
<tr>
<td>Deer</td>
<td>Deer (96.06%)</td>
</tr>
<tr>
<td>Dog</td>
<td>Dog (96.29%)</td>
</tr>
<tr>
<td>Frog</td>
<td>Frog (96.49%)</td>
</tr>
<tr>
<td>Horse</td>
<td>Horse (75.84%)</td>
</tr>
<tr>
<td>Ship</td>
<td>Ship (95.46%)</td>
</tr>
<tr>
<td>Truck</td>
<td>Truck (96.88%)</td>
</tr>
</tbody>
</table>

Figure 4. Histograms showing the distribution of confidences after bootstrapping for correct/incorrect classifications (left), and how the highest confidence samples are disproportionally clean (right) for 50% Symmetric Noise on CIFAR10.

4.6. Ablations and Training Time

In Tables 8, 9 and 10 we perform a number of ablation studies on the CIFAR10 40% Asymmetric noise. In Table 8 we show the accuracy of our trained model after each stage of training. We see that the model accuracy improves after each stage of training, with the final training’s use of MixUp and strong augmentations providing an additional 0.87% accuracy over the semi-supervised learning stage.

In Table 9 we see the large impact that the choice of augmentations has on the number of errors in the clean set after bootstrapping. We see that using strong augmentations for training greatly reduces the number of errors, likely due to their regularising effect and its ability to prevent overfitting to the noisy labels. During the evaluation stage however, averaging the prediction of the model over multiple weak augmentations performs best. We see that this matches the findings by Nishi et al. [34], who find that using strong augmentations for training and weaker augmentations for loss modelling works best.

In Table 10 we see the similarly large impact that using self-supervised pretraining has on the number of errors in the clean set after bootstrapping. In the supplementary material, we show additional experiments with ‘null labels’ and different model architectures.

Table 8. Model accuracy after each stage of training on CIFAR10 Asym. 40% noise.

<table>
<thead>
<tr>
<th>Training Stage</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>After Bootstrapping</td>
<td>91.41</td>
</tr>
<tr>
<td>After Semi-Supervised Learning</td>
<td>94.98</td>
</tr>
<tr>
<td>After Final Training</td>
<td>95.85</td>
</tr>
</tbody>
</table>

Table 9. Effect of different training/testing augmentations on the number of errors in a clean set of 10,000 samples selected after bootstrapping. Test performed on CIFAR10 Asym. 40% noise.

<table>
<thead>
<tr>
<th>Evaluation Aug.</th>
<th>None</th>
<th>Weak</th>
<th>Strong</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>579</td>
<td>265</td>
<td>28</td>
</tr>
<tr>
<td>None</td>
<td>361</td>
<td>56</td>
<td>21</td>
</tr>
<tr>
<td>Weak</td>
<td>346</td>
<td>300</td>
<td>31</td>
</tr>
<tr>
<td>Strong</td>
<td>369</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 10. Effect of self-supervision on the number of errors in a clean set of 10,000 samples selected after bootstrapping. Test performed on CIFAR10 Asym. 40% noise.

As for training time on CIFAR10 problems, our method takes on average 13.8h for SimCLR pretraining, 0.5h for bootstrapping, 7.5h for SSL, and 2.5h for final training (total of 24.3h) on an Nvidia RTX 2080. In comparison, DivideMix [28] takes on average 5h, and the more recent method of PropMix [10] can take up to 10h.

5. Conclusions

In this paper, we proposed a new method that predicts labels from images and their noisy labels. Unlike other methods, our training procedure does not require access to a clean set of data, which we achieve by introducing bootstrapping and a careful noise-based balancing procedure. By utilising the relationship between images, noisy labels and ‘clean’ labels to accurately relabel samples, we find that we can achieve SOTA results. By simply changing the model used, we further unify the noisy label learning and semi-supervised learning domains, resulting in a simplified architecture that can improve performance for challenging instance-dependent noisy-label tasks. Additionally, we find that by randomly replacing noisy labels with ‘null’ labels during training, we can remove the need for model distillation, allowing practitioners of our method to perform predictions with and without noisy labels.
References


[30] Junnan Li, Caiming Xiong, and Steven CH Hoi. Learn-


[55] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points


