

Contrastive Representations for Label Noise Require Fine-Tuning

Pierre Nodet^{1,2}, Vincent Lemaire¹,
Alexis Bondu¹, and Antoine Cornuéjols²

¹ Orange Labs, Paris & Lannion, France

² AgroParisTech, Paris, France

Abstract. In this paper we show that the combination of a Contrastive representation with a label noise-robust classification head requires fine-tuning the representation in order to achieve state-of-the-art performances. Since fine-tuned representations are shown to outperform frozen ones, one can conclude that noise-robust classification heads are indeed able to promote meaningful representations if provided with a suitable starting point. Experiments are conducted to draw a comprehensive picture of performances by featuring six methods and nine noise instances of three different kinds (none, symmetric, and asymmetric). In presence of noise the experiments show that fine tuning of Contrastive representation allows the six methods to achieve better results than end-to-end learning and represent a new reference compare to the recent state of art. Results are also remarkable stable versus the noise level.

1 Introduction

Deep Learning (DL) paradigm has proved very powerful in many tasks, however recent papers [34, 49] have shown that “**noisy labels**” are a real challenge for end-to-end deep learning architectures. Their test performance is found to deteriorate significantly even if they are able to learn perfectly the train examples. This problem has attracted a lot of suggestion in many recent papers.

Zhang et al. [50] conducted experiments to analyze the impact of label noise on deep architectures, and they found that the performance degradation mainly comes from the representation learning rather than the classification part. It therefore appears very difficult to learn a relevant representation in the presence of label noise, in an end-to-end manner.

To tackle this problem, one option is to exploit an already existing representation which has been learned in an unsupervised way. In particular, Self Supervised Learning [24] (SSL) gathers an ensemble of algorithms which automatically generate supervised tasks from unlabeled data, and, therefore to learn representations from examples that are not affected by label noise. An example of SSL algorithm is Contrastive Learning [21], where a representation of the data is learned by making feature vectors from similar pictures (i.e. generated from the same original picture by using two different transformer functions) to be close in the feature space whereas feature vectors from dissimilar pictures are to

be far apart. In [13], the authors propose to initialize the representation with a pre-trained Contrastive Learning one, and then, to use the noisy labels to learn the classification part and fine-tune the representation. It appears that this approach clearly outperforms the end-to-end architecture, where the representation is learned from noisy labels.

But questions remain: is this performance improvement only attributable to the quality of the Contrastive Representation used (i.e. the starting point of fine-tuning)? Or is the fine-tuning step able to promote a better representation? To answer these questions this paper examines the different possibilities to learn a DL architecture in presence of label noise: (i) end-to-end learning (ii) learning only the head part when freezing a contrastive representation and (iii) fine tuning the later representation.

The rest of this paper is organized as follows. The section 2 provides a brief overview of the main families of algorithms dedicated to fight the label noise underlying the issue of preserving a good representation in spite of label noise. Section 3 then describes the experimental protocol. The section 4 will present the results and a deep analysis which will allow us to answer the questions above. The last section raises an interesting conclusion and provides some perspectives for future work.

2 Representation Preserving with Noisy Labels

This section presents a brief overview of the state of the art on *learning deep architecture with noisy labels* emphasizing how these methods *preserve*, to some extent, *the learned representation* in the presence of label noise. For an extended overview, the reader may look [43].

2.1 Preserving by Recovering

The dominant approach to preserve the learned representation is to recover a clean distribution of the data from the noisy dataset. It mostly consists in finding a mapping function from the noisy to the clean distribution thanks to heuristics, algorithms or machine learning models. Three different ways of recovering the clean distribution are usually put forward [36]: (i) sample reweighting; (ii) label correction and (iii) instance moving.

Recovering by Reweighting - The sample reweighting methodology aims at assigning a weight to every samples such that the reweighted population behaves as being sampled from the clean distribution. The Radon-Nikodym derivative (RND) [35] of the clean concept with respect to the noisy concept is the function that defines the perfect reweighting scheme. Many algorithms therefore rely on providing a good estimation of the RND by learning it from the data using Meta Learning [38] or minimizing the Maximum Mean Discrepancy of both distributions in a Reproducing kernel Hilbert space [8, 32]. Many of these methods are inspired by the covariate shift problem [14, 20]. Other algorithms rely on different reweighting schemes that do not involve the RND as done, for instance,

in Curriculum Learning [2]. They are described in details later in this section. By doing sample reweighting, algorithms evaluate whether or not a sample is deemed to have been corrupted and assign a lower weight to a suspect sample so that its influence on the training procedure is lowered. The hope is that clean samples are sufficient to learn high-quality representations

Recovering by Relabelling - Another way to recover the clean distribution from the noisy data is to correct the noisy labels. One great advantage over sample reweighting is that corrected samples can be fully used during the training procedure. Indeed, when a sample is corrected, it will count as one entire sample in the training procedure (gradient descent for example), whereas a reweighted noisy sample would get a low weight and would not be used significantly in the training procedure. Thus, when done effectively, label correcting might get better performance. Meta Label Correction (MLC) [42] is an example of this approach where the label correction is done thanks to a model learned using meta learning. One downside of label correction, however, is that the label of a clean sample can get “corrected” or the label of a noisy sample can get changed to a wrong label. Label correcting algorithm assign the same weight to all training examples, even though they might have “corrected” a label based on shaky assumptions. By contrast, Sample reweighting will assign a low weight if the algorithm is not confident in whether the sample is clean or noisy.

Recovering by Modifying - A third way to recover the clean distribution is by modifying the sample itself so that its position in the feature space gets closer or is moved within an area for its label that seems more appropriate (i.e. obeying regularisation criteria). Finding a transformation in the latent space itself has the advantage to require less labelled samples, or even none at all, as the work is performed on distance between samples themselves, like for example in [40].

2.2 Preserving by Collaboration

Multiple algorithms and agreements measures have been used in many sub-fields of machine learning such as ensembling [4, 10, 11] or semi supervised learning [3, 48]. They can be adapted to learn with noisy labels by relying on a disagreement method between models in order to detect noisy samples. When the learned models disagree on predictions for the label of a sample, this is considered as a sign that the label of this sample may be noisy. When the models used are diverse enough, these methods are often found to be quite efficient [17, 46].

However these algorithms suffer from learning their own biases and diversity needs to be introduced in the learning procedure. Using algorithms from different classes of models and different origins can increase the diversity among them by introducing more source of biases [28]. Alternating between learning from the data and from the other models is another way to combat the reinforcement of the models’ biases [46]. These algorithms rely on carefully made heuristics to be efficient.

2.3 Preserving by Correcting

When learning loss base models, such as neural networks, on label noise, the loss value of a training example can be a discriminative feature to decide if its label is noisy. Deep neural networks seem to have the property that they first learn general and high level patterns from the data before falling prey to overfitting the training samples, especially in the presence of noisy labels [1, 30]. As they are “learned” at a later stage, these noisy examples are often associated with a high loss value [16] which may then highly influences the training procedure and perturb the learned representation [49]. A way to combat label noise is accordingly to focus first on small loss and easy examples and keep the high loss and hard examples for the end of the training procedure. Curriculum Learning [2] is a way to employ this training schema with heuristic based schemes [9, 22, 27, 31] or schemes learned from data [23, 41]. This class of algorithm has the same properties as the ones relying on importance reweighting, but maybe more adapted to training with iterative loss based algorithms such as neural networks or linear models.

Instead of filtering or reweighting samples based on their loss values, one could try to correct the loss for these samples using the underlying noise pattern. Numerous method have been doing so by estimating the noise transition matrix for *Completely at Random* (i.e uniform) and *At Random* (i.e class dependent) noise [19, 37, 42]. This category of algorithms are still to be tested on more complex noises scenarios such as *Not at Random* (i.e instance and class dependent) noise.

2.4 Preserving by Robustness

The last identified way to preserve the learned representation of a deep neural network in presence of label noise is by using a robust or regularized training procedure. This can take multiple forms from losses to architectures or even optimizers. One of them are Symmetric Losses [5, 12, 39]. A symmetric loss has the property that: $\forall x \in \mathcal{X}, \sum_{y \in \mathcal{Y}} L(f(x), y) = c$ where $c \in \mathbb{R}$. These losses have been proven to be theoretically insensitive to Completely at Random (CAR) label noise. Recently, modified versions of the well-known Categorical Cross Entropy (CCE) loss have been designed in order to be more robust and thus more resistant to CAR label noise as is the case for the Symmetric Cross Entropy (SCE) loss [45] or the Generalized Cross Entropy Loss (GCE) [51]. Both of these rely on using the CCE loss combined with a known more robust loss such as the Mean Absolute Error (MAE). However, the resulting algorithms often underfit in presence of too few label noise while they are unable to learn a correct classifier with too much label noise.

All these approaches still adopt the end-to-end learning framework, aiming at fighting the effects of label noise by preserving the learned representation. However they fail to do so in practice: *decoupling* the learning of the representation, using Self Supervised (SSL) learning, from the classification learning stage itself and then fine tuning the representation with robust algorithms is beneficial

for the model performance [13, 50]. A natural question arises about the origin of the performance improvements, and the ability of these algorithms to learn or promote a good representation in presence of label noise. If robust algorithms are unable to learn a representation it should be even better to freeze the SSL representation instead of fine tuning it.

In order to assess the origin of the improvements for different classes of algorithms and different noise levels, we compare the above-mentioned end-to-end approaches against each other when the representation is learned in a self-supervised fashion by either fine tuning or freezing the representation when the classification head is learned. Thus, any difference in the performance would be attributable to the difference in the representation learnt.

3 Experimental Protocol

In [50], the authors showed that when using end-to-end learning, fine tuning the representation on noisy labels harms a lot the final performance, while learning a classifier on frozen embeddings is quite robust to label noise and leads to significant performance improvements over state-of-the-art algorithms if the representation is learned using trustful examples. The latter can be found for instance using confidence and loss value. Nonetheless it is arguable whether these improvements were brought by an efficient self-supervised pretraining (SSL) with SimCLR [6], a contrastive learning method, or by the classification stage of the REED algorithm [50].

The goal of the following experimental protocol is to assess and isolate the role of the contrastive learning stage, in the performance that can be achieved by representative methods as presented in Section 2 about state of the art approaches. Specifically, several RLL algorithms have been chosen, one from each of the highlighted families (see Section 2 and Table 1). For each, the difference in performance between using contrastive learning to learn the representation and the performance reported with the original end-to-end algorithms is measured. These experiments seek to highlight the impact of each RLL algorithms and assess if these are able to promote a better representation than the pretrained contrastive representation through fine-tuning.

The rest of this section describes the experimental protocol used to conduct this set of experiments.

3.1 The tested Algorithms

Section 2 presented an overview of the state of the art for learning with label noise organized around families of approaches that we highlighted. Since our experiments aim at studying the properties of each of these approaches, we selected one representative technique from each of these families as indicated in the following.

- In the first family of techniques (*recover the clean distribution*), the algorithms re-weight the noisy examples or attempt to correct their label. One

of these algorithm uses what is called Dynamic Importance Reweighting (**DIW**). It reweights samples using Kernel Mean Matching (KMM) [14, 20] as is done in covariate shift with Density Ratio Estimators [44]. Because this algorithm adapts well-grounded principles to end-to-end deep learning, it is a particularly relevant algorithm for our experiments.

- CoLearning (**CoL**) [46] is a good representative of the family of *collaborative learning algorithms*. It uses disagreements criteria to detect noisy labels and is tailored for end-to-end deep learning where the two models are branches of a larger neural networks. It appears to be one of the best performing collaborative algorithm while not resorting to complex methods such as data augmentation or probabilistic modelling like the better known DivideMix [29].
- The third identified way to combat label noise is by *mitigating the effect of high loss samples* [16] by either ditching them or using a loss correction approach. Curriculum learning is often used to remove the examples that are associated with high loss from the training set. (**MWNet**) [41] is one the most recent approach using this technique, which learns the curriculum from the data with meta learning. Besides, Forward Loss Correction (**F-Correction**) [37] and Gold Loss Correction (**GLC**) [19] are two of the most popular approaches to combat label noise by *correcting the loss function*. Both seek to estimate the transition matrix between the noisy labels to the clean labels, the first technique using a supervised approach thanks to a clean validation set, and the second one in an unsupervised manner. Even though many extensions of these algorithm have been developed since then [42, 47], in these experiments, we use F-Correction and GLC since they are way simpler and almost as effective.
- Lastly, in recent literature, a new emphasis is put on the research of new loss functions that are conducive to better risk minimization in presence of noisy labels *for robustness purpose*. For example, [5, 39] show theoretically and experimentally that when the loss function satisfies a symmetry condition, described below, this contributes to the robustness of the classifier. The Generalized Cross Entropy (**GCE**) [51] is the robust loss chosen in this benchmark as it appears to be very effective.

A note about additional requirements: These algorithms may have additional requirements, mostly some knowledge about the noise properties. These are described in table 1. In the experiments presented below, the clean validation dataset is set to be 2 percent of the total training data, like in [41, 53], and the noise probability is provided to the algorithms that need it.

A note about the choice of the pretrained architecture: We chose to use SimCLR for Self-Supervised Learning (SSL) as done in [50].

SimCLR is a contrastive learning algorithm that is composed of three main components (See Figure 1): a family of data augmentation \mathcal{T} , an encoder network $f(\cdot)$ and a projection head $g(\cdot)$. Data augmentation is used as a mean to generate positive pairs of samples: a single image \mathbf{x} is transformed into two similar images

Algorithms (Date)	Noise Ratio	Clean Validation	Family (Section)
DIW (2020)	×	✓	Reweighting (2.1)
CoLearning (2020)	✓	×	Collaborative Learning (2.2)
MWNet (2019)	×	✓	Curriculum Learning (2.3)
F-Correction (2017)	×	×	Loss Correction (2.3)
GLC (2018)	×	✓	Loss Correction (2.3)
GCE (2018)	×	×	Robust Loss (2.4)

Table 1. Taxonomy of robust deep learning algorithms studied in this paper. The **Noise Ratio** column corresponds to whether the algorithm needs the noise rate (✓) to learn from noisy data or not (×). The **Clean Validation** column corresponds to whether the algorithm needs an additional clean validation dataset (✓) to learn from noisy data or not (×).

$\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ by using a data augmentation module \mathcal{T} with different seeds t and t' . Then the two images go through an encoder network $f(\cdot)$ to extract an image representation \mathbf{h} , such as $\mathbf{h}_i = f(\tilde{\mathbf{x}}_i)$ and $\mathbf{h}_j = f(\tilde{\mathbf{x}}_j)$. Finally a projection head $g(\cdot)$ is used to train the contrastive objective in a smaller sample space \mathbf{z} , with $\mathbf{z}_i = g(\tilde{\mathbf{h}}_i)$ and $\mathbf{z}_j = g(\tilde{\mathbf{h}}_j)$. The contrastive loss used is called the NT-Xent, the normalized temperature-scaled cross entropy loss, and defined by the following formula:

$$\ell(\mathbf{z}_i, \mathbf{z}_j) = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_j)/\tau)}{\sum_{k=1}^{2N} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)} \quad (1)$$

where τ is the temperature scaling and sim is the cosine similarity. The final loss is computed across all positive pairs, both (i, j) and (j, i) , in a mini-batch. When the training of SimCLR is complete, the projection head $g(\cdot)$ is dropped and the embeddings \mathbf{h} are used as an image representation in downstream tasks.

Other SSL algorithms could have been used as well, such as Moco [7, 18] or Bootstrap Your Own Latent (BYOL) [15]. However, we do not expect that the main conclusions of the study would be much changed.

3.2 Datasets

The datasets chosen in this benchmark are two image classification datasets namely CIFAR10, CIFAR100. They are two famous image classification datasets, containing only clean examples and as such, we will simulate symmetric (Completely at Random) and asymmetric (At Random) noise as defined later in section 3.3. These benchmarks should be extended to other image classification datasets such as FashionMNIST, Food-101N, Clothing1M and Webvision and to other classification tasks such as text classification or time series classification.

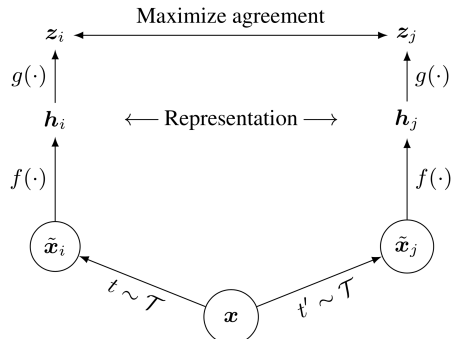


Fig. 1. Figure from [6]: “A simple framework for contrastive learning of visual representations. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training is completed, we throw away the projection head $g(\cdot)$ and use encoder $f(\cdot)$ and representation \mathbf{h} for downstream tasks.”

3.3 Simulated Noise

As datasets chosen in Section 3.2 contains clean labels, label noise will be introduced synthetically on the training samples. Two artificial noise models will be used, a symmetric (Completely at Random) and asymmetric (At Random) noise. Symmetric noise corrupts a label from one class to any other classes with the same probability, meanwhile the asymmetric corrupts a label to a similar class only. Similar classes are defined through class mappings. For CIFAR-10, the class mappings are TRUCK \rightarrow AUTOMOBILE, BIRD \rightarrow AIRPLANE, DEER \rightarrow HORSE, CAT \leftrightarrow DOG. For CIFAR-100, the class mappings are generated from the next class in that group (where 100 classes are categorized into 20 super-classes of 5 classes). These class mappings are the ones introduced in [37, 51].

3.4 Implementation Details

We give some implementation details for reproducibility and / or a better understanding of the freezing process in the experiments:

- On CIFAR10 and CIFAR100 the SGD optimizer will be used to train the final Multinomial Logistic Regression with an initial learning rate of 0.01, a weight decay of $1e^{-4}$ and a non-Nesterov momentum of 0.9. The learning rate will be modified during training with cosine annealing [33]. The batch size is 128.
- When doing the “Freeze” experiments, the weights of SimCLR from [13] will be used and will not be modified during the training procedure. All

the weights up to before the projection head of SimCLR are used, then the dimension output of the feature encoder is 2048 for CIFAR10 and CIFAR100. The classification architecture is composed by a single linear layer with an output dimension of 10 (or 100), corresponding to the number of classes. Thus when trained with the Categorical Cross Entropy it corresponds to a usual logistic regression. This classifier is going to be learned with multiple algorithms robust to label noise. These algorithms are not modified from their original formulation.

- The "*Fine Tuning*" experiments follow the same implementation as the "*Freeze*" experiments. However the weights of the same pretrained SimCLR encoder are allowed to be modified by backpropagation.
- Based on their public implementation and / or article we re-implemented all the algorithm tested (DIW [8], CoL [46], MWNet [41], F-Correction [37], GLC [19] and GCE [51]). All these re-implemented algorithms will soon be available as an open source library easily usable by researchers and practitioners. These custom implementations have been verified to produce, under the same condition stated in the corresponding original papers (noise models, network architectures, optimizers, ...), the same results or results in the interval of confidence (for clean or noisy labels). We may thus be confident that results in the different parts of the Tables 2 and 3 are comparable.
- The experiments have been run multiple times for all algorithms, some datasets, some noise models and some noise ratios with different seeds to see the seed impact on the final performance of the classifier. For all algorithms, the standard deviation of the accuracy was less than 0.1 percent.

4 Results

This section reports the results obtained using the protocol described in section 3. They are presented in the tables 2 and 3 corresponding to the two tested datasets CIFAR10 and CIFAR100. Each table is composed of four rows subsections corresponding to the different types of representation used, which can be learned in a *End-to-End* manner (A), be taken from an already existing SSL model, either *Frozen* (B) or *Fine tuned* (C). Moreover they are composed of two columns subsections corresponding to the noise model used to corrupt samples (symmetric or asymmetric).

These tables present the results from different studies: (A) The first part of these tables about "*End-to-End learning*" are results reported in the respective papers [8, 19, 37, 41, 46, 51] or reported in [13]; (B) The second part about "*Freeze*" experiments conducted in this paper, are made by re-implementing the referred algorithms from scratch; (C) The "*Fine Tuning*" experiments are results reported in [13].

The interpretation of the Table 2 and 3 will be done in two times, first a comparison between whole blocks (as (A) against (B)) will give insights on how deep neural networks learn representations on noisy data and how robust algorithms helps to improve the learning process or helps to preserve a given

Algorithms		CIFAR10								
		Clean		Symmetric					Asymmetric	
		0	20	40	60	80	90	95	20	40
DIW [8]	End-to-End (A)			80.4	76.3					84.4
CoL [46]			93.3	91.2		49.2			88.2	82.9
MWNet [41]		95.6	92.4	89.3	84.1	69.6	25.8	18.5	93.1	89.7
F-Correction [37]		90.5	87.9			63.3	42.9		90.1	
GLC [19]		95.0	95.0	95.0	95.0	90.0	80.0	76.0		
GCE [51]		93.3	89.8	87.1	82.5	64.1			89.3	76.7
DIW	Freeze (B)	91.3	91.2	90.8	90.5	89.8	89.2	88.1	91.0	90.6
CoL		91.1	91.1	90.9	90.6	89.9	89.4	88.8	90.8	89.9
MWNet		91.3	91.2	90.8	90.6	89.8	88.2	82.4	90.9	86.4
F-Correction		90.8	90.5	90.1	89.6	88.4	88.0	88.1	88.9	88.4
GLC		90.7	89.7	90.0	89.5	89.0	88.5	88.3	88.7	88.2
GCE		91.1	90.8	90.7	90.5	90.4	90.0	89.1	90.9	89.0
DIW	Fine Tuning (C)	94.5	94.5	94.5	94.5	94.0	92.0	89.1	94.2	93.6
CoL		93.9	94.6	94.6	94.2	93.6	92.7	91.7	94.0	93.7
MWNet [13]		94.6	93.9	92.9	91.5	90.2	87.2	93.7	92.6	
F-Correction		94.0	93.4	93.1	92.9	92.3	91.4	90.0	93.6	92.8
GLC		93.5	93.4	93.5	93.1	92.0	91.2	88.3	93.2	92.1
GCE [13]		94.6	94.0	92.9	90.8	88.4	83.8	93.5	90.3	

Table 2. Final accuracy for the different models on CIFAR10 under symmetric and asymmetric noises and multiple noise rates.

representation. Then in a second time comparisons in a given block will be made against multiple algorithms to see how well these conclusions works on different preservation families given in Section 2.

First, we observe when comparing section (A) and (B) from both tables that *Freeze* experiments consistently outperforms *End-to-End* experiments as soon as the data stop being perfectly clean. Using a pretrained self-supervised representation such as SimCLR improves significantly the performances of the final classifier. Outside of well controlled and perfectly clean datasets all selected algorithms are not able to learn a good enough representation from the noisy data and are beaten by a representation learned without resorting to using given labels. Robust Learning to Label noise algorithms, especially designed for deep learning, can preserve an already good representation from noisy labels but are unable to learn a good representation from scratch.

Then, we observe when comparing section (B) and (C) from both tables that *Fine Tuning* experiments consistently outperforms *Freeze* at noise rates less than 80 for the symmetric case and less than 40 for the asymmetric case. The nature of the final classifier used after the learned representation partially explains these results; we used a single dense layer (see Section 3.4). This classifier may under-fit as the number of learnable parameters might be too low to actually fit complex datasets such as CIFAR10 and CIFAR100 even with a good given representation. Using more complex classifiers such as Multi-Layer Perceptron could have led to comparable performances than fine tuning even for low noise rates. This point leaves room for further investigation. Having the possibility to

Algorithms		CIFAR100								
		Clean		Symmetric					Asymmetric	
		0	20	40	60	80	90	95	20	40
DIW [8]	End-to-End (A)			53.7	49.1					54.0
CoL [46]			75.8	73.0		32.8				
MWNet [41]		79.9	74.0	67.7	58.7	30.5	5.2	3.0	71.5	56.0
F-Correction [37]		68.1	58.6			19.9	10.2		64.2	
GLC [19]		75.0	75.0	75.0	62.0	44.0	24.0	12.0	75.0	75.0
GCE [51]		76.8	66.8	61.8	53.2	29.2			66.6	47.2
DIW	Freeze (B)	65.6	65.1	64.0	62.9	59.0	53.3	42.5	61.7	49.0
CoL		65.8	65.0	64.0	63.4	62.3	60.0	57.0	64.1	58.6
MWNet		66.6	66.6	66.2	65.4	63.7	59.8	49.5	64.8	54.5
F-Correction		66.5	64.7	61.8	58.8	54.5	51.7	50.8	58.4	56.5
GLC		58.5	57.8	52.3	51.1	41.6	40.1	35.3	51.4	50.3
GCE		63.5	62.9	61.5	60.0	55.7	51.0	49.9	51.2	48.3
DIW	Fine Tuning (C)	73.8	74.9	74.9	74.5	70.2	62.3	50.4	71.8	62.8
CoL		73.7	74.8	74.8	75.0	73.2	67.3	62.0	72.6	70.3
MWNet [13]		75.4	73.2	69.9	64.0	57.6	44.9	72.2	64.9	
F-Correction		69.8	70.1	69.1	69.5	66.9	62.1	57.0	70.3	66.2
GLC		69.7	69.4	68.6	62.5	50.4	32.1	18.7	68.2	62.3
GCE [13]		75.4	73.3	70.1	63.3	55.9	45.7	71.3	59.3	

Table 3. Final accuracy for the different models on CIFAR100 under symmetric and asymmetric noises and multiple noise rates.

fine tune the representation to better fit the classification task induces the risk to actually degrade it.

Outside of well controlled and perfectly clean datasets, practitioners should first consider to learn a self-supervised representation and then either fine tune it or freeze it with classifier learned with robust algorithms. Self-Supervised Learning (SSL) algorithm such as SimCLR seems to perfectly fit this task, but other SSL algorithms could be used and explored.

Another observation from this benchmark is about the difference in performance between all the tested algorithms. Indeed, if we consider part (B) of Table 2, for both noise models and all noise rates, the performances between the algorithms are close, around 0.1 point in accuracy with some exceptional data points. It shows that even complex algorithms have a hard time beating simpler approaches when they are compared with an already learned representation.

The same observation can be done for the part (B) of Table 3 (for CIFAR 100), especially for the symmetric noise. However the differences between algorithms are better put in perspective with this more complex dataset which contains 10 time more classes and 10 time less samples per classes. We notice that some algorithms start to struggle at high symmetric noise rate or for the more complex asymmetric noise model. For example, GLC is under-performing against competitors for all cases and is under-performing against its end-to-end version. One reason could be the small size used for the validation dataset as the transition matrix is evaluated on it in a supervised manner. The small number of samples may impact the performance of the transition matrix estimator. Much less so than the estimator proposed by F-Correction which seems to perform fine

even on CIFAR100 for all symmetric noises, yet only above average on asymmetric noises. Seeing F-Correction and GLC not performing well on asymmetric noise for both dataset is surprising as these algorithms were both particularly designed for this case.

Lastly we observe on both Tables 2 and 3 that algorithms with additional knowledge on the noise model (see Table 1) have an edge over algorithms that do not, especially on the hardest cases with more classes, higher noise ratio or more complex noise model. CoL requires the noise ratio as its efficiency relies on the hyper parameters value corresponding to the injection of pseudo labels and confidence in model prediction that are dependent of the noise ratio. CoL emerges among the most well rounded and most efficient algorithm for all noise models, noise rates and datasets thanks partially to this additional knowledge. On the other hand, GLC, DIW and MWNet require an additional clean validation dataset in order to estimate the noise model or a proxy of it to correct the learning procedure on the noisy dataset. We could expect these algorithms to perform better than CoL as they would be able to deal with more complex noise models and have a fine-grained policy for correcting noisy samples. Still these algorithms are not able in these experiments to get a better accuracy than CoL and perform on par with it.

Finally we need to emphasize that only two datasets have been used in this study, specially two datasets about image classification. In order to stronger our claims, more experiments should be conducted.

5 Conclusion

In this paper our contribution was to suggest new insights about *decoupling* against *end-to-end* deep learning architectures to learn, preserve or promote a good representation in case of label noise. We presented (i) a new view on a part of the state of the art: the ways to preserve the representation (ii) and an empirical study which completes the results and the conclusions of other recent papers [13, 50, 52]. Experiments conducted draw a comprehensive picture of performances by featuring six methods and nine noise instances of three different kinds (none, symmetric, and asymmetric). Our added value for the empirical study is the comparison between the "freeze" and the "fine tuning" results.

One conclusion we are able to draw is that designing algorithms that preserve or promote good representation under label noise is not the same as designing algorithms capable of learning from scratch a good representation under label noise. To make end-to-end learning succeed in this setup researchers should take a better approach when designing such algorithms.

Another element that emerged from the experiments was the efficiency of both freeze and fine tuning approaches in comparison to the end-to-end learning approach. Even the most complex algorithms such as DIW when trained in an end-to-end manner are not able to beat simple robust loss as GCE when trained with fine tuning. It questions usual experimental protocols of Robust Learning to Label (RLL) noise papers and questions the recent advances in the field.

Evaluating RLL algorithms with pretrained architectures should be the norm as it is easy to do so and the most efficient way for practitioners to train model on noisy data.

One more strong point in this conclusion is that in presence of noise the experiments show that fine tuning of Contrastive representation allows the six methods to achieve better results than their end-to-end learning version and represent a new reference compare to the recent state of art. Results are also remarkable stable versus the noise level.

Since fine-tuned representations are shown to outperform frozen ones, one can conclude that noise-robust classification heads are indeed able to promote meaningful representations if provided with a suitable starting point (contrastingly to readers of [13, 52] who might prematurely jump to the inverse conclusion).

However these experiments could be extended to be more exhaustive in two ways: (i) SimCLR is not the only recent and efficient contrastive learning algorithms, MOCO [7, 18] or Bootstrap Your Own Latent (BYOL) [15] could have been used as said earlier in the paper, but other self-supervised or unsupervised algorithms could have been used such as Auto-Encoder [26] or Flow [25]; (ii) experiments could be extended to datasets from other domains such as text classification or time series classification.

Acknowledgements

We would like to thank the anonymous reviewers for their careful, valuable and constructive reviews as well as the words of encouragement on our manuscript.

References

1. Arpit, D., Jastrzebski, S., Ballas, N., Krueger, D., Bengio, E., Kanwal, M.S., Maharaj, T., Fischer, A., Courville, A., Bengio, Y., Lacoste-Julien, S.: A closer look at memorization in deep networks. In: International Conference on Machine Learning. pp. 233–242 (2017)
2. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: International Conference on Machine Learning. pp. 41–48 (2009)
3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the eleventh annual conference on Computational learning theory. pp. 92–100 (1998)
4. Breiman, L.: Bagging predictors. *Machine Language* **24**(2), 123–140 (Aug 1996)
5. Charoenphakdee, N., Lee, J., Sugiyama, M.: On symmetric losses for learning from corrupted labels. In: International Conference on Machine Learning. vol. 97, pp. 961–970 (2019)
6. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 1597–1607. PMLR (13–18 Jul 2020)
7. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv:2003.04297 (2020)

8. Fang, T., Lu, N., Niu, G., Sugiyama, M.: Rethinking importance weighting for deep learning under distribution shift. In: *Neural Information Processing Systems* (2020)
9. Felzenszwalb, P., McAllester, D., Ramanan, D.: A discriminatively trained, multi-scale, deformable part model. In: *2008 IEEE conference on Computer Vision and Pattern Recognition*. pp. 1–8. IEEE (2008)
10. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* **55**(1), 119–139 (1997)
11. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. *Annals of Statistics* **29**, 1189–1232 (2001)
12. Ghosh, A., Kumar, H., Sastry, P.S.: Robust loss functions under label noise for deep neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence* **31**(1) (2017)
13. Ghosh, A., Lan, A.: Contrastive learning improves model robustness under label noise. *arXiv:2104.08984 [cs.LG]* (2021)
14. Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., Schölkopf, B.: Covariate shift by kernel mean matching. *Dataset shift in machine learning* **3**(4), 5 (2009)
15. Grill, J.B., Strub, F., Alché, F., Tallec, C., Richemond, P., Buchatskaya, E., Dörsch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., Kavukcuoglu, k., Munos, R., Valko, M.: Bootstrap your own latent - a new approach to self-supervised learning. In: *Advances in Neural Information Processing Systems*. vol. 33, pp. 21271–21284 (2020)
16. Gui, X.J., Wang, W., Tian, Z.H.: Towards understanding deep learning from noisy labels with small-loss criterion. In: *International Joint Conference on Artificial Intelligence* (2021)
17. Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., Sugiyama, M.: Co-teaching: Robust training of deep neural networks with extremely noisy labels p. 11 (2018)
18. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2020)
19. Hendrycks, D., Mazeika, M., Wilson, D., Gimpel, K.: Using trusted data to train deep networks on labels corrupted by severe noise. In: *Advances in Neural Information Processing Systems*. vol. 31, pp. 10456–10465 (2018)
20. Huang, J., Gretton, A., Borgwardt, K., Schölkopf, B., Smola, A.J.: Correcting sample selection bias by unlabeled data. In: *Advances in neural information processing systems*. pp. 601–608 (2007)
21. Jaiswal, A., Babu, A.R., Zadeh, M.Z., Banerjee, D., Makedon, F.: A survey on contrastive self-supervised learning. *Technologies* **9**(1), 2 (2021)
22. Jiang, L., Meng, D., Zhao, Q., Shan, S., Hauptmann, A.: Self-paced curriculum learning. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 29 (2015)
23. Jiang, L., Zhou, Z., Leung, T., Li, L.J., Fei-Fei, L.: MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In: *International Conference on Machine Learning*. vol. 80, pp. 2304–2313 (2018)
24. Jing, L., Tian, Y.: Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020)

25. Kobyzev, I., Prince, S., Brubaker, M.: Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020)
26. Kramer, M.A.: Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal* **37**(2), 233–243 (Feb 1991)
27. Kumar, M., Packer, B., Koller, D.: Self-paced learning for latent variable models. In: *Advances in Neural Information Processing Systems*. vol. 23 (2010)
28. Lee, J., Chung, S.Y.: Robust training with ensemble consensus. In: *International Conference on Learning Representations* (2020)
29. Li, J., Socher, R., Hoi, S.C.: Dividemix: Learning with noisy labels as semi-supervised learning. In: *International Conference on Learning Representations* (2019)
30. Li, M., Soltanolkotabi, M., Oymak, S.: Gradient descent with early stopping is provably robust to label noise for overparameterized neural networks. In: *International Conference on Artificial Intelligence and Statistics*. pp. 4313–4324 (2020)
31. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: *Proceedings of the IEEE international conference on computer vision*. pp. 2980–2988 (2017)
32. Liu, T., Tao, D.: Classification with noisy labels by importance reweighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**(3), 447–461 (Mar 2016)
33. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016)
34. Maennel, H., Alabdulmohsin, I.M., Tolstikhin, I.O., Baldock, R.J.N., Bousquet, O., Gelly, S., Keysers, D.: What do neural networks learn when trained with random labels? In: *Neural Information Processing Systems* (2020)
35. Nikodym, O.: Sur une généralisation des intégrales de m. j. radon. *Fundamenta Mathematicae* **15**(1), 131–179 (1930)
36. Nodet, P., Lemaire, V., Bondu, A., Cornuéjols, A.: Importance reweighting for biquality learning. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN)* (2021)
37. Patrini, G., Rozza, A., Menon, A., Nock, R., Qu, L.: Making deep neural networks robust to label noise: a loss correction approach. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017)
38. Ren, M., Zeng, W., Yang, B., Urtasun, R.: Learning to reweight examples for robust deep learning. In: *International Conference on Machine Learning*. vol. 80, pp. 4334–4343 (2018)
39. van Rooyen, B., Menon, A., Williamson, R.C.: Learning with symmetric label noise: The importance of being unhinged. In: *Neural Information Processing Systems*, pp. 10–18 (2015)
40. Shi, Y., Sha, F.: Information-theoretical learning of discriminative clusters for unsupervised domain adaptation. In: *ICML* (2012)
41. Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., Meng, D.: Meta-weight-net: Learning an explicit mapping for sample weighting. In: *Neural Information Processing Systems*. vol. 32 (2019)
42. Shu, J., Zhao, Q., Xu, Z., Meng, D.: Meta transition adaptation for robust deep learning with noisy labels. *arXiv:2006.05697* (2020)
43. Song, H., Kim, M., Park, D., Shin, Y., Lee, J.G.: Learning from noisy labels with deep neural networks: A survey. *arXiv:2007.08199 [cs.LG]* (2021)
44. Sugiyama, M., Suzuki, T., Kanamori, T.: Density ratio estimation: A comprehensive review (statistical experiment and its related topics) (2010)

45. Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., Bailey, J.: Symmetric cross entropy for robust learning with noisy labels. In: IEEE/CVF International Conference on Computer Vision. pp. 322–330 (2019)
46. Wang, Y., Huang, R., Huang, G., Song, S., Wu, C.: Collaborative learning with corrupted labels. *Neural Networks* **125**, 205–213 (2020)
47. Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., Sugiyama, M.: Are anchor points really indispensable in label-noise learning? In: *NeurIPS* (2019)
48. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: 33rd annual meeting of the association for computational linguistics. pp. 189–196 (1995)
49. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM* **64**(3), 107–115 (2021)
50. Zhang, H., Yao, Q.: Decoupling representation and classifier for noisy label learning. *arXiv:2011.08145* (2020)
51. Zhang, Z., Sabuncu, M.: Generalized cross entropy loss for training deep neural networks with noisy labels. In: *Neural Information Processing Systems*. vol. 31 (2018)
52. Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A.M., Litany, O.: Contrast to divide: Self-supervised pre-training for learning with noisy labels. *arXiv:2103.13646 [cs.LG]* (2021)
53. Zheng, G., Awadallah, A.H., Dumais, S.: Meta label correction for noisy label learning. *Proceedings of the AAAI Conference on Artificial Intelligence* **35** (2021)