

Article

A Framework Using Contrastive Learning for Classification with Noisy Labels

Madalina Ciortan ^{*,†}, Romain Dupuis [†]  and Thomas Peel

R&D Department, EURA NOVA, 1435 Mont-Saint-Guibert, Belgium; romain.dupuis@euranova.eu (R.D.); thomas.peel@euranova.eu (T.P.)

* Correspondence: madalina.ciortan@euranova.eu

† These authors contributed equally to this work.

Abstract: We propose a framework using contrastive learning as a pre-training task to perform image classification in the presence of noisy labels. Recent strategies, such as pseudo-labeling, sample selection with Gaussian Mixture models, and weighted supervised contrastive learning have, been combined into a fine-tuning phase following the pre-training. In this paper, we provide an extensive empirical study showing that a preliminary contrastive learning step brings a significant gain in performance when using different loss functions: non robust, robust, and early-learning regularized. Our experiments performed on standard benchmarks and real-world datasets demonstrate that: (i) the contrastive pre-training increases the robustness of any loss function to noisy labels and (ii) the additional fine-tuning phase can further improve accuracy, but at the cost of additional complexity.

Keywords: noisy labels; image classification; contrastive learning; robust loss



Citation: Ciortan, M.; Dupuis, R.; Peel, T. A Framework Using Contrastive Learning for Classification with Noisy Labels. *Data* **2021**, *6*, 61. <https://doi.org/10.3390/data6060061>

Academic Editors: M. Rashedur Rahman, Ken Barker, Luis Paulo F. Garcia, Nabeel Mohammed and Sifat Momen

Received: 29 April 2021

Accepted: 5 June 2021

Published: 9 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Collecting large and well-annotated datasets for image classification tasks represents a challenge as human quality annotations are expensive and time-consuming. Alternative methods exist, such as web crawlers [1]. Nevertheless, these methods generate noisy labels decreasing the performance of deep neural networks. They tend to overfit to noisy labels due to their high capacity [2]. That is why developing efficient noisy-label learning (NLL) techniques is of great importance.

Various strategies have been proposed to deal with NLL: (i) Noise transition matrix [3–5] estimates the noise probability and corrects the loss function, (ii) a small and clean subset can help to avoid overfitting [6], (iii) sample selection identifies true-labeled samples [7–9], and (iv) robust loss functions solve the classification problem only by adapting the loss function to be less sensitive to noisy labels [10–12]. Methods also combine other strategies (e.g., ELR+ [9], DivideMix [13]): two networks, semi-supervised learning, label correction, or mixup. They show the most promising results but lead to a large number of hyperparameters. That is why we explore improvement strategies for robust loss functions. They are simpler to integrate and faster to train, but as illustrated in Figure 1, they tend to overfit and have lower performance for high noise ratios.

Meanwhile, new self-supervised contrastive learning algorithms for image representations have been recently developed [14,15]. Such algorithms extract representation (or features) in unsupervised settings by comparing among the input samples. These representations can then be used for downstream tasks such as classification. For this task, methods based on contrastive learning and fine-tuned only on a small fraction of all available labels compete with fully supervised learning. Therefore, using contrastive learning to extract features independently of the amount of noise for NLL appears as promising.

This work is a wide experimental exercise analyzing the current state-of-the-art in noisy label image classification and evaluating if recent (unsupervised) contrastive learning

techniques can also be used to provide robustness to classifiers trained in a noisy setting with non-robust and robust loss functions. The key contributions of our work are:

- A framework increasing robustness of any loss function to noisy labels by adding a contrastive pre-training task.
- The adaptation of the supervised contrastive loss to use sample weight values, representing the probability of correctness for each sample in the training set
- An extensive empirical study identifying and benchmarking additional state-of-the-art strategies to boost the performance of pre-trained models: pseudo-labeling, sample selection with GMM, weighted supervised contrastive learning, and mixup with bootstrapping.

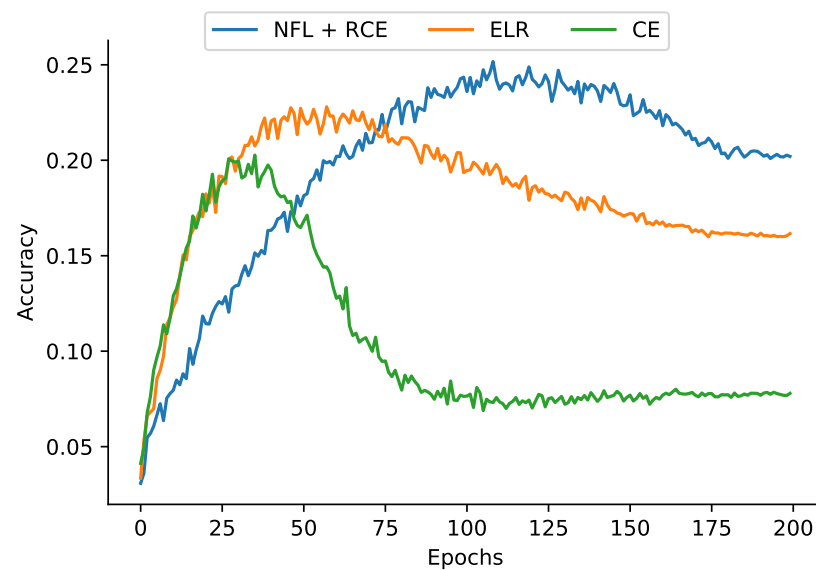


Figure 1. Top-1 test accuracy for a ResNet18 trained on the CIFAR-100 dataset with a symmetric noise of 80% for three losses: Cross Entropy (CE), Normalized Focal Loss + Reverse Cross Entropy (NFL+RCE), and Early Learning Regularization (ELR).

2. Related Works

Existing approaches dealing with NLL and contrastive learning in computer vision are briefly reviewed. Extra details can be found in Song et al. [16], Le-Khac et al. [17].

2.1. Noise Tolerant Classification

Sample Selection: This method identifies noisy and clean samples within the training data. Several strategies leverage the interactions between multiple networks to identify the probably correct labels [7–9]. Recent works [18,19] exploit the small loss trick to identify clean and noisy samples by considering a certain number of small-loss training samples as true-labeled samples. This approach can be justified by the memorization effect: deep neural networks first fit the training data with clean labels during a so-called early learning phase, before overfitting the noisy samples during the memorization phase [13,20].

Robust Loss Function: Commonly used loss functions, such as Cross Entropy (CE) or Focal Loss, are not robust to noisy labels. Therefore, new loss functions have been designed. Such robust loss functions can be easily incorporated into existing pipelines to improve performance regarding noisy labels. The symmetric Cross Entropy [11] has been proposed by adding a reverse CE loss to the initial CE. This combination improves the accuracy of the model compared to classical loss functions. Ma et al. [12] show theoretically that normalization can convert classical loss functions into loss functions robust to noise labels. The combination of two robust loss functions can also improve robustness. However,

the performance of normalized loss functions remains quite low for high noise rates, as illustrated in Figure 1.

Semi-supervised: Semi-supervised approaches deal with both labeled and unlabeled data. Recent works [9,21,22] combine sample selection with semi-supervised methods: the possibly noisy samples are treated as unlabeled and the possibly clean samples are treated as labeled. Such approaches leverage information contained in noisy data, for instance by using MixMatch [23]. Semi-supervised approaches show competitive results. However, they use several hyperparameters that can be sensitive to changes in data or noise type [16,24].

Contrastive learning: recent developments in self-supervised and contrastive learning [24–26] inspire new approaches in NLL. Li et al. [26] employed features learned by contrastive learning to detect out-of-distribution samples.

2.2. Contrastive Learning for Vision Data

Contrastive learning extracts features by comparing each data sample with different samples. The central idea is to bring different instances of the same input image closer and spread instances from different images apart. The inputs are usually divided into positive (similar inputs) or negative pairs (dissimilar inputs). Data augmentation is typically used to create positive pairs. Frameworks have been recently developed, such as CPCv2 [27], SimCLR [14], Moco [15]. Once the self-supervised model is trained, the extracted representations can be used for downstream tasks. In this work, the representations are used for noisy label classification.

Chen et al. [14] demonstrate that large sets of negatives (and large batches) are crucial in learning good representations. However, large batches are limited by GPU memory. Maintaining a memory bank accumulating a large number of negative representations is an elegant solution decoupling the batch size from the number of negatives [28]. Nevertheless, the representations get outdated in a few iterations. The Momentum Encoder [15] addresses the issues by generating a dynamic memory queue of representations. Other strategies aim at getting more meaningful negative samples to reduce the memory/batch size [29].

Recent methods in contrastive learning have been developed to avoid computing pairwise comparisons with negative samples. The Bootstrap Your Own Latent (BYOL) algorithm [30] does not use any negative samples and compares only positive pairs with a momentum encoder (similar to Moco) and a stop-gradient mechanism. This mechanism is also used in the SimSiam method [31]. Caron et al. [32] introduce a cluster algorithm and enforce consistency between cluster assignments for different augmentations instead of a direct pairwise comparison.

3. Preliminaries

Let $D = \{(\mathbf{x}_i, \bar{y}_i)\}_{i=1..n}$, $\mathbf{x}_i \in \mathbb{R}^{d_1 \times d_2}$, $\bar{y}_i \in \{1, \dots, K\}$ denote a noisy input dataset with an unknown number of samples incorrectly labeled. The associated true and unobservable labels are written as y_i . The images \mathbf{x}_i are of size $d_1 \times d_2$ and the classification problem has K classes. The goal is to train a deep neural network (DNN) f . Using a robust loss function for training consists of minimizing the empirical risk defined by robust loss functions in order to find the set of optimal parameters θ . The one-hot encoding of the label is denoted by the distribution $q(k|\mathbf{x})$ for a sample \mathbf{x} and a class k , such as $q(y_i|\mathbf{x}_i) = 1$ and $q(k \neq y_i|\mathbf{x}_i) = 0$, $\forall i \in \{1, \dots, n\}$. The probability vector of f is defined by the softmax function $p(k|\mathbf{x}) = \frac{e^{z_k}}{\sum_{j=1}^K e^{z_j}}$ where z_k denotes the logits output with respect to class k .

3.1. Classification with Robust Loss Functions

Our method employs noise-robust losses to train the classifier in the presence of noisy labels. Such losses improve the classification accuracy compared to the commonly used Cross Entropy (CE), as illustrated in Figure 1. In this section, the general empirical risk for a given mini-batch is defined by $L = \sum_{i=1}^N \mathcal{L}(f(x_i), \bar{y}_i) = \sum_{i=1}^N l_i$. The term l_i is modified by each loss function.

The classical CE is used as a baseline loss function not robust to noisy labels [33] and is defined as:

$$l_{ce} = - \sum_{k=1}^K q(k|\mathbf{x}_i) \log(p(k|\mathbf{x}_i)). \quad (1)$$

As presented in Section 2, Ma et al. [12] introduce robust loss functions called Active Passive Losses that do not suffer from underfitting. We investigate the combination between the Normalized Focal Loss (NFL) and the Reversed Cross Entropy (RCE) called NFL+RCE. It shows promising results on various benchmarks. The NFL is defined as:

$$l_{nfl} = \frac{- \sum_{k=1}^K q(k|\mathbf{x}_i) (1 - p(k|\mathbf{x}_i))^\gamma \log(p(k|\mathbf{x}_i))}{- \sum_{j=1}^K \sum_{k=1}^K q(y = j|\mathbf{x}_i) (1 - p(k|\mathbf{x}_i))^\gamma \log(p(k|\mathbf{x}_i))}, \quad (2)$$

where $\gamma \geq 0$ is a hyperparameter. The RCE loss is:

$$l_{rce} = - \sum_{k=1}^K p(k|\mathbf{x}_i) \log(q(k|\mathbf{x}_i)). \quad (3)$$

The final combination following the framework simply gives a different α and β to each loss:

$$l_{nfl+rce} = \alpha \cdot l_{nfl} + \beta \cdot l_{rce}. \quad (4)$$

The two hyperparameters α and β control the balancing between more active learning and less passive learning. For simplicity, α and β are set to 1.0 without any tuning.

Liu et al. [13] propose another framework to deal with noisy annotations based on the “early learning” phase. The loss, called Early Learning Regularization (ELR), adds a regularization term to capitalize on early learning. ELR is not, strictly speaking, a robust loss but belongs to robust penalization and label correction methods. The penalization term corrects the CE based on estimated soft labels identified with semi-supervised learning techniques. It prevents memorization of false labels by steering the model towards these targets. The regularization term maximizes the inner product between model outputs and targets:

$$l_{elr} = l_{ce} + \frac{\lambda_{elr}}{N} \log \left(1 - \sum_{k=1}^K p(k|\mathbf{x}_i) t(k|\mathbf{x}_i) \right). \quad (5)$$

The target is not set equal to the model output but is estimated with a temporal ensembling from semi-supervised methods. Let $t(k|\mathbf{x}_i)^{(l)}$ denote the target for example \mathbf{x}_i at iteration l of training with a momentum β :

$$t(k|\mathbf{x}_i)^{(l)} = \beta t(k|\mathbf{x}_i)^{(l-1)} + (1 - \beta) p(k|\mathbf{x}_i)^{(l)}. \quad (6)$$

A more complex method improving accuracy, called ELR+, has also been developed and combines weight averaging, two parallel networks, and a mixup data augmentation.

The importance in the choice of the loss function is also discussed for other applications, such as image segmentation [34] or survival data [35].

3.2. Contrastive Learning

Contrastive learning methods learn representations by contrasting positive and negative examples. A typical framework is composed of several blocks [36]:

- Data augmentation: Data augmentation is used to decouple the pretext tasks from the network architecture. Chen et al. [14] study broadly the impact of data augmentation. We follow their suggestion combining random crop (and flip), color distortion, Gaussian blur, and gray-scaling.
- Encoding: The encoder extracts features (or representation) from augmented data samples. A classical choice for the encoder is the ResNet model [37] for image data. The final goal of the contrastive approach is to find correct weights for the encoder.
- Loss function: The loss function usually combines positive and negative pairs. The Noise Contrastive Estimation (NCE) and its variants are popular choices. The general formulation for such a loss function is defined for the i th pair as [38]:

$$L_i = -\log \frac{\exp(\mathbf{z}_i^T \mathbf{z}_{j(i)}/\tau)}{\sum_{a \in A(i)} \exp(\mathbf{z}_i^T \mathbf{z}_a/\tau)}, \text{ with } i \in I, \quad (7)$$

where \mathbf{z} is a feature vector, I is the set of indexes in the mini-batch, i is the index of the anchor, $j(i)$ is the index of an augmented version of the anchor source image, $A(i) = I \setminus \{i\}$, and τ is a temperature controlling the dot product. The denominator includes one positive and K negative pairs. The temperature has two competing effects on the loss function: low temperatures help the model to learn from hard negatives while high temperatures allow to use larger learning rate making the optimization easier, and classes more separated [14,39].

- Projection head: That step is not used in all frameworks. The projection head maps the representation to a lower-dimensional space and acts as an intermediate layer between the representation and the embedding pairs. Chen et al. [14,31] show that the projection head helps to improve the representation quality.

4. A Framework Coupling Contrastive Learning and Noisy Labels

As illustrated in Figure 2, our method classifies noisy samples in a two-phased process. First, a classifier pre-trained with contrastive learning produces train-set pseudo-labels (pre-training phase, in panel a), used during the training of a subsequent fine-tuning phase (panel b). The underlying intuition is that the predicted pseudo-labels are more accurate than the original noisy labels. The contrastive learning step performed in the first phase (panel a1) is supposed to reduce the label noise sensitivity of the classifier (panel a2) thanks to the computed representations; the resulting model can be also used in a standalone way with a reduced number of hyperparameters, without the underlying fine-tuning phase.

The second phase leverages the pseudo-labels predicted by the pre-training in all underlying steps (b1–b3). To mitigate the effect of potentially incorrectly predicted pseudo-labels, a Gaussian Mixture Model (GMM, panel b1) with 2 components follows the small loss-trick to predict for each sample the probability of correctness. This value is used as a weight in a supervised contrastive step (panel b2), performed to improve the learned representations by taking advantage of the label information. A classification head is added to the contrastive model in order to produce the final predictions (panel b3). The fine-tuning phase can be seen as an adaptation of the pre-training phase to handle pseudo-labels.

To maximize the impact of the contrastive learning on the underlying classification, the supervised training is performed in 2 steps: a warm-up step, updating only the classifier layer (while keeping the encoder frozen) is followed by the full model training. We compared three different loss functions for the supervised classification: the classical CE, the robust NFL+RCE, and the ELR loss.

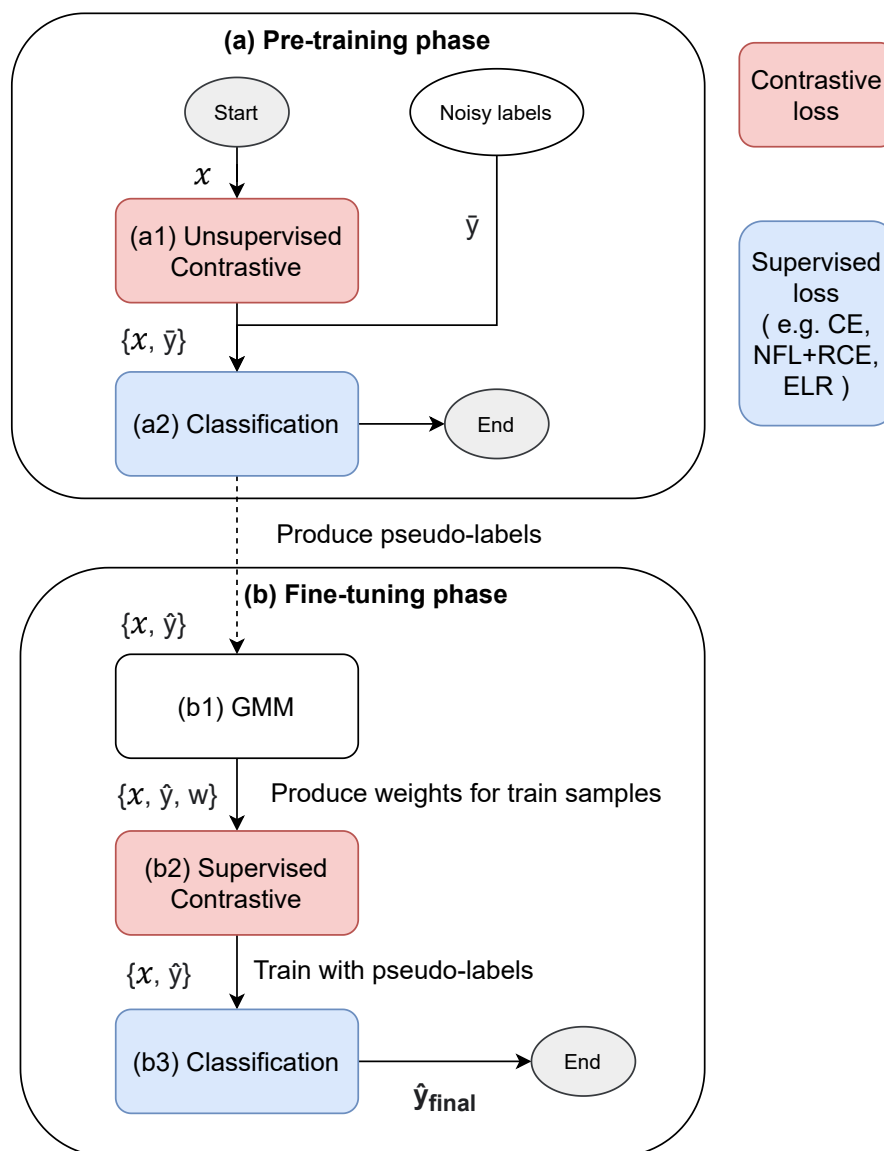


Figure 2. Overview of the framework consisting of two phases: pre-training (panel a) and fine-tuning (panel b). After a contrastive learning phase (a1), a classifier (a2) is trained to predict train-set pseudo-labels \hat{y} . The fine-tuning phase uses \hat{y} as a new ground truth. First, a GMM model (b1) predicts the probability of correctness for each sample, used as a corrective weight factor in a supervised contrastive training (panel b2). The final predictions \hat{y}_{final} are produced by the (b3) classifier.

4.1. Sample Selection and Correction with Pseudo-Labels

Pseudo labels represent one hot encoded model's predictions on the training set. Pseudo-labels were initially used in semi-supervised learning to produce annotations for unlabeled data; in the noisy label setting, various techniques (e.g., DivideMix) identify a subset with a high likelihood of correctness and treat the remaining samples as the unlabeled counterpart in semi-supervised learning. In this work, we elaborate on the observation that the training set labels, predicted after training the model with a noise-robust loss function (i.e., the pseudo labels), are more accurate than the ground truth. This observation is supported by the results in Figure 3, depicting the accuracy of pseudo labels predicted on CIFAR100, contaminated with various levels of asymmetric (panel a) and symmetric (panel b) noise. The pseudo labels are more accurate than the corrupted ground truth in both settings and bring a higher gain in performance as the noise ratio increases.

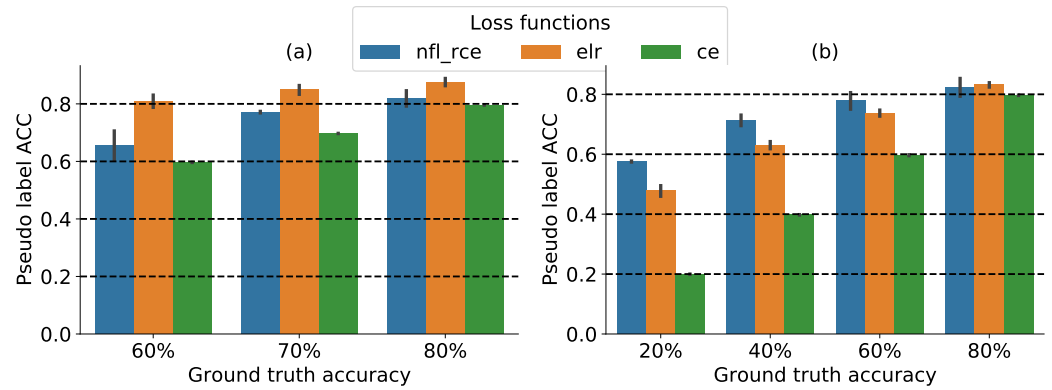


Figure 3. Accuracy of pseudo labels on all simulated settings with asymmetric (a) and symmetric (b) noise, evaluated on CIFAR100. The correctness of the ground truth is represented on the x-axis, while the accuracy of predicted pseudo labels on the y-axis. In all experiments, the pseudo labels have a higher accuracy than the corrupted ground truth and this gain increases with the noise ratio.

As proposed by other approaches [18], the loss value on train samples can be used to discriminate between clean and mislabeled samples. The sample correctness probability is computed by fitting a 2-components GMM on the distribution of losses [9]. The underlying probability is used as a sample weight:

$$w_i = p(k = 0 | l_i), \quad (8)$$

where l_i is the loss for sample i and $k = 0$ is the GMM component associated to the clean samples (lowest loss). Figure 4 depicts the evolution of the clean training set identified by GMM on an example: its accuracy grows from 0.6 to 0.93 while the size stabilizes at 60% of the training set.

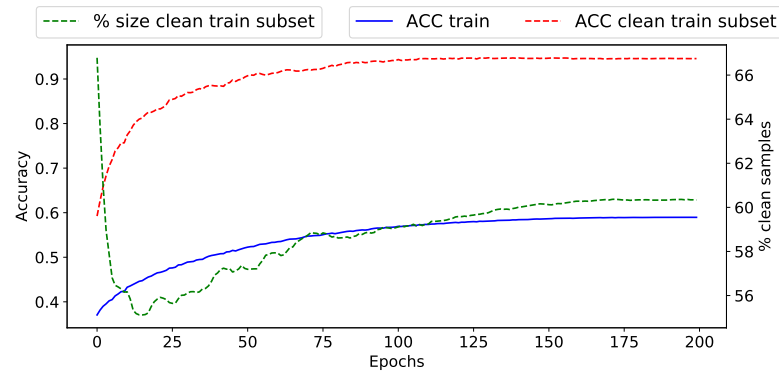


Figure 4. Accuracy of the entire training set (in blue) compared to the clean train subset (in red); the clean subset's percentual size is depicted in green. The example is performed on CIFAR100, with 40% symmetric noise.

4.2. Weighted Supervised Contrastive Learning

A modification to the contrastive loss defined in Equation (7) has been proposed to leverage label information [39]:

$$L_i = -\log \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\exp(\mathbf{z}_i^T \mathbf{z}_p / \tau)}{\sum_{a \in A(i)} \exp(\mathbf{z}_i^T \mathbf{z}_a / \tau)}, \quad (9)$$

where $P(i) = \{j \in I \setminus \{i\}, y_j = \tilde{y}_i\}$ with \tilde{y}_i the prediction of the model for input x_i .

As explained in the previous section, the loss value for the training set sample is used to fit a GMM with 2 components, corresponding to correctly and incorrectly labeled

samples. We adapted the supervised representation loss to employ w , a weighting factor representing the sample probability of membership to the correctly labeled component. Thus, likely mislabeled samples having large loss values would contribute only marginally to the supervised representations:

$$L_i = -\log \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\widetilde{w}_{p,i} \exp(\mathbf{z}_i^T \mathbf{z}_p / \tau)}{\sum_{a \in A(i)} \exp(\mathbf{z}_i^T \mathbf{z}_p / \tau)}, \quad (10)$$

where $\widetilde{w}_{p,i}$ is a modified version of w_p such as $\widetilde{w}_{p,i} = 1$ if $p = j(i)$ else $\widetilde{w}_{p,i} = w_i$. If all samples are considered as noisy, Equation (10) is simplified into the classical unsupervised contrastive loss in Equation (7).

5. Experiments

The framework is assessed on three benchmarks and the contribution of each block identified in Figure 2 is analyzed.

5.1. Datasets

CIFAR10 and CIFAR100 [40]. These experiments assess the accuracy of the method against synthetic label noise. The two datasets are contaminated with simulated symmetric or asymmetric label noise reproducing the heuristic in Ma et al. [12]. The symmetric noise consists of corrupting an equal arbitrary ratio of labels for each class. The noise level varies from 0.2 to 0.8. For asymmetric noise [3,13], sample labels have been flipped within a specific set of classes, thus providing confusion between predetermined pairs of labels. For CIFAR100, 20 groups of super-classes have been created, each consisting of 5 sub-classes. The label flipping is performed only within each super-class circularly. The asymmetric noise ratio is explored between 0.2 and 0.4.

Webvision [41]. This is a real-world dataset with noisy labels. It contains 2.4 million images crawled from the web (Google and Flickr) that share the same 1000 classes from the ImageNet dataset. The noise ratio varies from 0.5% to 88%, depending on the class. In order to speed-up the training time, we used mini Webvision [7], consisting of only the top 50 classes in the Google subset (66,000 images).

Clothing1M [42]. Clothing1M is a large real-world dataset consisting of 1 million images on 14 classes of clothing articles. Being gathered from e-commerce websites, Clothing1M embeds an unknown ratio of label noise. Additional validation and test sets, consisting of 14 k and 10 k clean labeled samples, have been made available. In order to speed-up the training time, we selected a subset of 56,000 images keeping the initial class distribution.

Both Webvision and Clothing1M images were resized to 128×128 . Therefore, the reported results may differ from other papers cropping the images to a 224×224 resolution. Numerical details about the different datasets can be found in Appendix A.

5.2. Settings

We use the contrastive SimCLR framework [14] (<https://github.com/HobbitLong/SupContrast> (accessed on 14 December 2020)) with a ResNet18 [37] (without ImageNet pre-training) as the encoder. A projection head was added after the encoder for the contrastive learning (of dimension 128 for CIFAR and dimension 512 for Webvision and Clothing1M) with the following architecture: a multi-layer perceptron with one hidden layer and a ReLU non-linearity. The classifier following the contrastive learning step has a simple multi-layer architecture: a single hidden layer with batch normalization and a ReLU activation function. A comparison with a linear classifier is provided in Appendix C.3.

For all supervised classification, we use the SGD optimizer with momentum 0.9 and cosine learning rate annealing. The NFL hyperparameter γ is set to 0.5. Unlike the original paper, the ELR hyperparameters do not depend on the noise type: the regularization

coefficient λ_{elr} and the momentum β are set to 3.0 and 0.7. Details on the experiment setting can be found in Appendix B.

All codes are implemented in the PyTorch framework [43]. The experiments for CIFAR are performed with a single Nvidia TITAN V-12GB and the experiments for Webvision and Clothing1M are performed with a single Nvidia Tesla V100-32GB, demonstrating the accessibility of the method. Our implementation has been made available along with the Supplementary Materials.

6. Results

All experiments presented in this section evaluate our method's performance with the top-1 accuracy score.

6.1. Impact of Contrastive Pre-Training

To evaluate the impact of the contrastive pre-training on the classification model, the proposed method (pre-training phase) is compared with a baseline classifier, trained for 200 epochs without contrastive learning. For each simulated dataset, we compare robust losses (e.g., NLF+RCE and ELR) and Cross Entropy. Results for CIFAR10 and CIFAR100 are depicted in Table 1 for different levels of symmetric and asymmetric noise. The pre-training improves the accuracy of the three different baselines for both datasets with different types and ratios of label noise. The largest differences are observed for the noisiest case with 80% noise. The pre-training outperforms the baselines by large margins between 10 and 75 for CIFAR10 and between 5 and 30 for CIFAR100.

In addition to the comparisons with ELR and NLF+RCE, performed using our implementations (column Base in Table 1), we present the results reported by other recent competing methods. As shown in the Introduction, numerous contributions have been made to the field in the last years. Six recent representative methods are selected for comparison: Taks [44], Co-teaching+ [45], ELR [13], DivideMix [9], SELF [21], and JoCoR [46]. The results are presented in Table 2. The difference between the scores reported by ELR and those obtained with our run (using the same implementation, but slightly different hyperparameters and a ResNet18 instead of a ResNet34) suggests that the method is less stable on data contaminated with asymmetric noise and sensitive to small changes hyperparameters. Moreover, ELR proposes hyperparameters having different values depending on the type of dataset (i.e., CIFAR10/CIFAR100) and underlying noise (i.e., symmetric/asymmetric), identified after a hyperparameter search exercise. The best scores are reported by DivideMix and they surpass all other techniques. One can note DivideMix uses a PreAct ResNet18 while we use a classical ResNet18. Moreover, a recent study [24] attempted to replicate these values and reported significantly lower results on CIFAR100 (i.e., 49.5% instead of 59.6% on symmetric data and 50.9% instead of 72.1% on asymmetric data). Our framework compares favorably with the other competing methods, both on symmetric and asymmetric noise.

Webvision and Clothing1M results are presented in Table 3. The contrastive framework outperforms the respective baselines for the three loss functions. As a result that the images have a reduced size, and for Clothing1M, we use a smaller training set, the direct comparison with competing methods is less relevant. However, the observed gap in performance is significant and promising for training images with higher resolution. Moreover, a ResNet50 model has been trained with our framework on the Webvision dataset with a higher resolution (224×224). The accuracy reaches 75.7% and 76.2% for CE and ELR, respectively. These results are very close to the values reported with DivideMix (77.3%) and ELR+ (77.8%) using a larger model, Inception-ResNet-v2 (the difference is more than 4% on the ImageNet benchmark [47]).

Table 1. Results on both CIFAR10 and CIFAR100 using symmetric noise (0.2–0.8) and asymmetric noise (0.2–0.4). We compare training from scratch or from pre-trained representation. Best scores are in bold for each noise scenario and each loss.

Type	η	Loss	CIFAR10		CIFAR100	
			Base	Pre-t.	Base	Pre-t.
Sym	0.2	ce	77.2	87.7	55.6	56.5
		elr	90.3	93.0	64.1	67.4
		nfl+rce	91.0	92.7	66.6	68.8
	0.4	ce	58.2	78.0	39.9	41.9
		elr	82.3	92.0	56.9	62.0
		nfl+rce	87.0	91.4	60.2	66.3
	0.6	ce	35.2	59.2	21.8	26.8
		elr	64.2	90.4	40.6	55.7
		nfl+rce	80.2	88.1	47.0	61.8
	0.8	ce	17.0	27.3	7.80	12.4
		elr	18.3	84.8	16.2	45.3
		nfl+rce	42.8	59.9	20.1	50.2
Asym	0.2	ce	84.0	87.9	59.0	57.8
		elr	91.8	92.4	70.3	70.2
		nfl+rce	90.2	91.5	63.9	68.4
	0.3	ce	79.2	83.9	50.6	50.4
		elr	89.6	91.7	69.8	69.3
		nfl+rce	86.7	89.9	53.5	63.5
	0.4	ce	75.3	77.8	41.8	42.4
		elr	72.3	89.5	67.6	67.6
		nfl+rce	80.0	82.4	40.6	47.8

Table 2. Accuracy scores compared with 6 methods (Taks, Co-teaching+, ELR, DivideMix, SELF, and JoCoR) on CIFAR10 (C10) and CIFAR100 (C100). The cases most affected by dropout are presented, with symmetric (S) and asymmetric (A) noise. Top-2 scores are in bold.

	C10 80% S	C10 40% A	C100 80% S	C100 40% A
Ours (ELR)	84.8	89.5	45.3	67.6
ELR [13]	73.9	91.1	29.7	73.2
Taks [44]	40.2	73.4	16.0	35.2
Co-teach+ [45]	23.5	68.5	14.0	34.3
DivideMix [9]	92.9	93.4	59.6	72.1
SELF [21]	69.9	89.1	42.1	53.8
JoCoR [46]	25.5	76.1	12.9	32.3

Table 3. Top-1 accuracy for mini-Webvision and Clothing1M. Best scores are in bold for each dataset and each loss. Pre-t represents the pre-training phase while Fine-tune refers to the results after the fine-tuning step.

Loss	Webvision			Clothing1M		
	Base.	Pre-t.	Fine-Tune	Base.	Pre-t.	Fine-Tune
ce	51.8	57.1	58.4	54.8	59.1	61.5
elr	53.0	58.1	59.0	57.4	60.8	60.4
nfl+rce	49.9	54.8	58.2	57.4	59.4	60.1

Supported by this first set of experiments, the preliminary pre-training with contrastive learning shows great performances. The accuracy of both traditional and robust-loss classification models is significantly improved.

6.2. Sensitivity to the Hyperparameters

Estimating the best hyperparameters is complex for datasets with noisy labels as clean validation sets are not available. For instance, Ortego et al. [24] show that two efficient methods (e.g., ELR and DivideMix) could be sensitive to specific hyperparameters. Therefore, a hyperparameter sensitivity study has been carried out to estimate the stability of the framework for the learning rate. Figure 5 depicts the sensitivity on CIFAR100 with 80% noise. CE and NFL+RCE seem to have opposite behaviors. The CE reaches competitive results with small learning rates but tends to overfitting for higher learning rates. The NFL+RCE loss tends to underfitting for the lowest learning rates but is quite robust for higher values. The ELR loss has the smallest sensitivity to the learning for the investigated range but does not reach the best values obtained with CE or NFL+RCE. We can assume that the regularization term coupled with pre-training is very efficient. It prevents memorization of the false labels as observed with CE. Results for other noise ratios are documented in Appendix C.2.

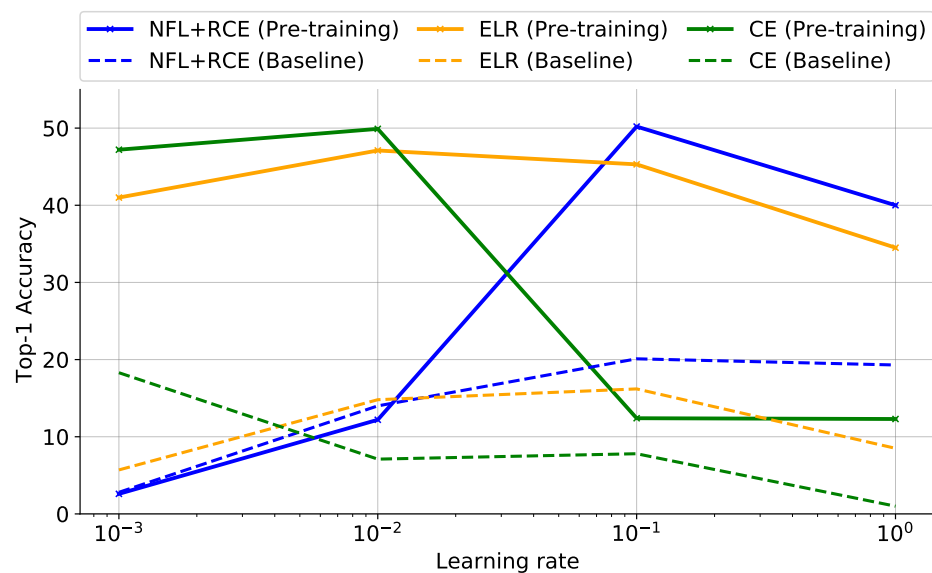


Figure 5. Learning rate sensitivity for CIFAR100 with 80% noise. The explored learning rate values are {0.001, 0.01, 0.1, 1.0}. The baseline (dashed line) is compared with our framework (solid line).

This sensitivity analysis is limited to the learning rate. Investigating the impact of other hyperparameters, such as the momentum β or the regularization factor λ_{elr} , could be interesting. In their original papers, ELR and NFL+RCE reach respectively 25.2% and 30.3% with other hyperparameters. These values are still far from the improvements brought by the contrastive pre-training but it suggests that the results could be improved with different hyperparameters.

Our empirical results indicate that the analyzed methods may be sensitive to hyperparameters. Despite the promised robustness to label noise, the analyzed robust losses are also affected by overfitting or underfitting. Our experiments have been built upon the parameters recommended in each issuing paper (e.g., ELR, SIMCLR) but, since the individual building blocks can be affected by small variations in input parameters, the performance of our method may also be impacted. Finding a relevant method to estimate proper hyperparameters in NLL remains a challenge.

As overfitting remains an important and recurrent issue affecting the training of noisy label models, we explored three strategies to identify when overfitting starts, without using a clean validation set. This constraint was imposed to simulate the real-world scenario, when only access to a potentially noise-corrupted dataset is provided. First, the behavior of a corrupted validation set was analyzed on symmetric and asymmetric datasets, but we failed to establish a consistent relation between the start of overfitting and the trend in the accuracy of the validation set. The accuracy of the validation set increases for the first few training epochs and then stabilizes over a plateau phase. Secondly, we explored a recent contribution, attempting to identify the Training Stop Point (TSP) [48] under similar conditions, but our experimental tests have shown that proposed heuristic does not consistently apply to all studied types of noise. Thirdly, we studied the Centered Kernel Alignment (CKA) [49] characterizing the representations created in the final network layers, but we could not identify a change in stability associated with the start of overfitting. Thus, we conclude that in the absence of a clean validation set, identifying when overfitting starts remains an open challenge for noisy labels. The three summarized experiments are detailed in Appendix G.

6.3. Impact of the Fine-Tuning Phase

Experimental results on synthetic label noise, depicted in Figure 6, show that continuing the presented pre-training block (Figure 2) with the fine-tuning phase increases the accuracy in over 65% of cases on CIFAR10 and over 80% of cases on CIFAR100. For both datasets, asymmetric noise data benefit more from this approach than symmetric noise. All experiments only use the input parameters proposed in the loss-issuing papers.

The sample selection has also got a positive impact on the two real-world datasets, as shown in Table 3 by the “Fine-tune” columns. The average accuracy improvement is about 1.8%. Only the ELR loss function slightly decreases the performance on Clothing1M.

Enriching pre-trained models with sample weighting and selection, pseudo labels instead of corrupted targets, and supervised contrastive pre-training can improve the classification accuracy. However, such an approach raises the question of a trade-off between complexity, accuracy improvement, and computation time.

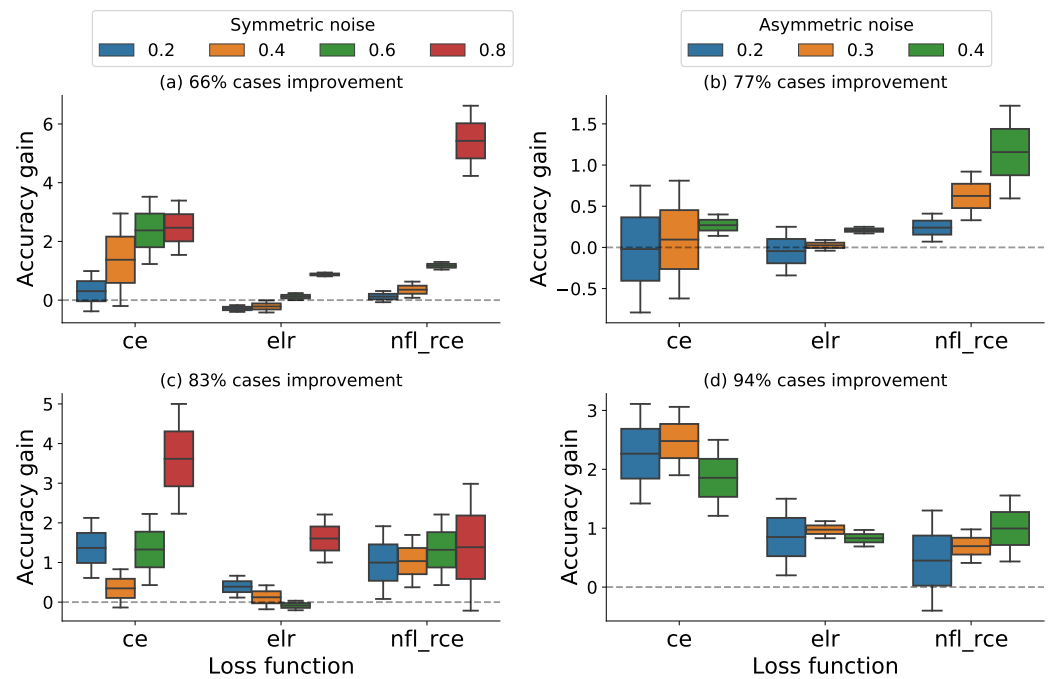


Figure 6. Accuracy gain when performing the fine-tuning phase after the pre-training block (computed as the difference between fine-tuning accuracy and pre-training accuracy). The plot gathers the results for all noise ratios on CIFAR10 (panels a,b) and CIFAR100 (c,d) with symmetric (first column) and asymmetric (second column) noise.

7. Discussion and Limits of the Framework

In addition to the presented fine-tuning phase, we evaluated the performance of other promising techniques, such as the dynamic bootstrapping with mixup [18]. This strategy has been developed to help convergence under extreme label noise conditions. Details can be found in Appendix D. The improvement that dynamic bootstrapping can bring when used after pre-training is depicted in Figure 7. In most of the cases, this technique improves the accuracy, as indicated by the positive accuracy gain scores, measuring the difference between the accuracy after dynamic bootstrapping and the accuracy of the pre-training phase. ELR and CE benefit most from this addition for CIFAR100. The impact of the dynamic bootstrapping should also be analyzed for the fine-tuning phase and for larger datasets, such as Webvision or Clothing1M.

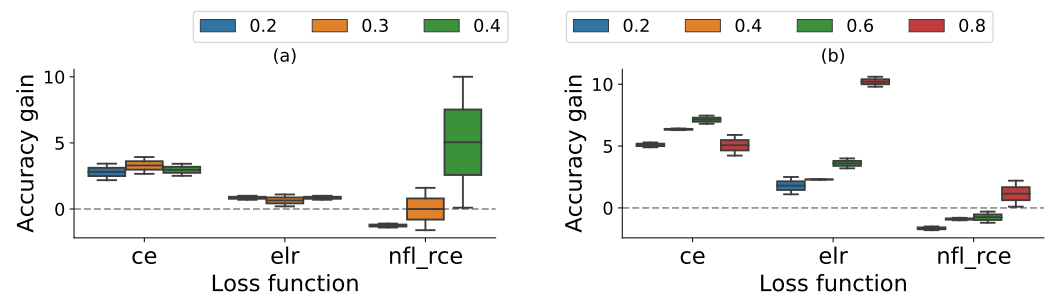


Figure 7. Top-1 accuracy gain for the dynamic bootstrapping on CIFAR100 with asymmetric (a) and symmetric noise (b). Dynamic bootstrapping is an alternative to the proposed fine-tuning phase. Each color is associated to a noise ratio.

One of the major drawbacks of our method is the extra computational time needed to learn representations with contrastive learning. A detailed study, comparing the execution time of our framework with 6 other competing methods has been provided in Appendix F. The pre-training phase doubles the execution time of a reference baseline, consisting of

performing only a single classification step, while the entire framework increases the execution time 3 to 4 times the baseline value. However, the contrastive learning does not increase the need for GPU memory if the batch size is limited for the contrastive learning [15,50]. The computational time could be reduced by initializing the contrastive step with the pre-trained weights from ImageNet.

Most state-of-the-art approaches also leverage computationally expensive settings, consisting of larger models (e.g., ResNet50), dual model training, or data augmentation such as mixup. In this work, we explored the limits of a restricted computational setting, consisting of a single GPU and 8GB RAM. All experiments use a ResNet18 model, batch sizes of 256, and for real-world datasets, the images have been rescaled (e.g., 128×128 instead of 224×224). We also foresee that the contrastive learning step could be improved by images with higher resolutions as smaller details could be identified in the representation embedding.

There remain multiple open problems for future research, such as: (i) identifying the start of the memorization phase in the absence of a clean dataset, (ii) studying the impact of contrastive learning on other models for noisy labels such as DivideMix, (iii) comparing SimCLR approach in the context of noisy labels with other contrastive frameworks (the impact of Moco is studied in Appendix C.1) and other self-supervised approaches, and (iv) having a better theoretical understanding of the interaction between the initial state pre-computed with contrastive learning and the classifier in presence of noisy labels. Moreover, the analysis carried out in this work should be validated on larger settings, in particular on Clothing1M with a ResNet50, higher resolutions, and the full dataset.

8. Conclusions

In this work, we presented a contrastive learning framework optimized with several adaptations for noisy label classification. Supported by an extensive range of experiments, we conclude that a preliminary representation pre-training improves the performance of both traditional and robust-loss classification models. Additionally, multiple techniques can be used to fine-tune and further optimize these results; however, no approach provides a significant improvement systematically on all types of datasets and label noise. The Cross Entropy penalized by Early Learning Regularization (ELR) shows the best overall results for synthetic noise but also real-world datasets.

However, the training phases remain sensitive to input configuration. Overfitting is the common weakness of all studied models. When trained with tuned parameters, even traditional (Cross Entropy) models provide competitive results, while robust-losses are less sensitive. The typical noisy label adaptations, such as sample selection or weighting, the usage of pseudo labels, or supervised contrastive losses, improve the performance to a lesser extent but increase the framework's complexity. We hope that this work will promote the use of contrastive learning to improve the robustness of the classification process with noisy labels.

Supplementary Materials: The archive called `python_scripts.zip` is available online at <https://www.mdpi.com/2306-5729/6/6/61/s1>. Our implementation has been made available along with this archive. It contains information about the environment, Python scripts, and Jupyter notebooks to run our experiments.

Author Contributions: Conceptualization, M.C., R.D., and T.P.; methodology, M.C., R.D., and T.P.; software, M.C. and R.D.; validation, M.C. and R.D.; formal analysis, M.C., R.D., and T.P.; investigation, M.C. and R.D.; data curation, M.C.; writing—original draft preparation, M.C. and R.D.; writing—review and editing, M.C., R.D., and T.P.; visualization, M.C. and R.D.; supervision, T.P.; project administration, T.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by a CooTech subsidy of the Wallonia Region via project Asterion convention number 8015. Same applies for article processing charge (APC) fees.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. CIFAR data can be found here: <https://www.cs.toronto.edu/~kriz/cifar.html> (accessed on 15 June 2020) and Webvision data can be found here: <https://data.vision.ee.ethz.ch/cv1/webvision/dataset2017.html> (accessed on 12 October 2020). Clothing1M data was obtained from a third party and are available from the authors by checking https://github.com/Cysu/noisy_label (accessed on 9 November 2020).

Conflicts of Interest: The authors declare no conflict of interest.

Sample Availability: Our implementation and models have been made available along with the supplementary materials.

Abbreviations

The following abbreviations are used in this manuscript:

NLL	Noisy-label learning
CE	Cross Entropy
NFL	Normalized Focal Loss
RCE	Reversed Cross Entropy
ELR	Early Learning Regularization
GMM	Gaussian Mixture Model

Appendix A. Description of the Datasets

Table A1 gives a detailed description of the datasets, including size of the training and test sets, the image resolution, and the number of classes.

Table A1. Description of the datasets used in the experiments.

Data Set	Train	Test	Size	# Classes
CIFAR10	50 K	10 K	32 × 32	10
CIFAR100	50 K	10 K	32 × 32	100
Clothing1M	56 K	5 K	128 × 128	14
mini Webvision	66 K	2.5 K	128 × 128	50

Appendix B. Detailed Settings of the Experiments

All experiments use the ResNet18 as an encoder. The classification steps are combined with data augmentation: a random crop with a padding of 4, a horizontal flip with a probability of 50%, and a random rotation of 20°. All other hyperparameters are shown in Table A2.

Table A2. Training parameters. Symbols: l.r. means learning rate, w.d. means weight decay, opti. means optimizer, Proj. dim. means the dimension of the projection head, Repre. means representation step, and Classi. means the supervised classification step.

	C10/C100	Webvision	Clothing1M
Batch	512	512	512
Opti.	Adam	Adam	Adam
Repre. l.r.	10 ⁻³	10 ⁻³	10 ⁻³
w.d.	10 ⁻⁶	10 ⁻⁶	10 ⁻⁶
Epochs	500	500	500
Proj. dim.	128	512	512

Table A2. *Cont.*

		C10/C100	Webvision	Clothing1M
Classi.	Batch	256	256	256
	Opti.	SGD	SGD	SGD
	l.r.	0.01/0.1	0.4	0.01
	w.d.	10^{-5}	3.10^{-5}	10^{-4}
	Epochs	200	200	200

Appendix C. Ablation Study

Appendix C.1. Contrastive Learning with a Momentum Encoder

The momentum encoder from the Moco framework [15] maintains a dynamic memory queue of representations. The current mini-batch is added to the memory queue while the oldest mini-batch is dequeued. The offline momentum encoder is a copy of the online encoder by taking an exponentially-weighted average of the parameter of the online encoder. The main advantage of Moco is to be able to reduce the batch size (and the GPU memory) while keeping a very large number of negative pairs for the contrastive learning.

The different representations computed by SimCLR and Moco are compared on CIFAR100. Both approaches are trained for 500 epochs following the usual hyperparameter parameters from the initial papers. As the two methods use different strategies to compute the representations, their quality is assessed by learning a linear classifier on top of the frozen encoder network. It can be seen as a proxy for representation quality. The SimCLR framework reaches 55.3% of accuracy while Moco gets 55.0% of accuracy. However, the two encoders do not behave in a similar way with regard to noisy labels. The same classifier (multi-layer, same learning rate and weight decay) is trained starting from the representation computed by SimCLR and Moco. As depicted in Table A3, the representations computed by Moco are more sensitive to the noisy labels. However, reducing the learning rate of the optimizer by a factor 10 (column Moco-Fine Tune) significantly increases the accuracy.

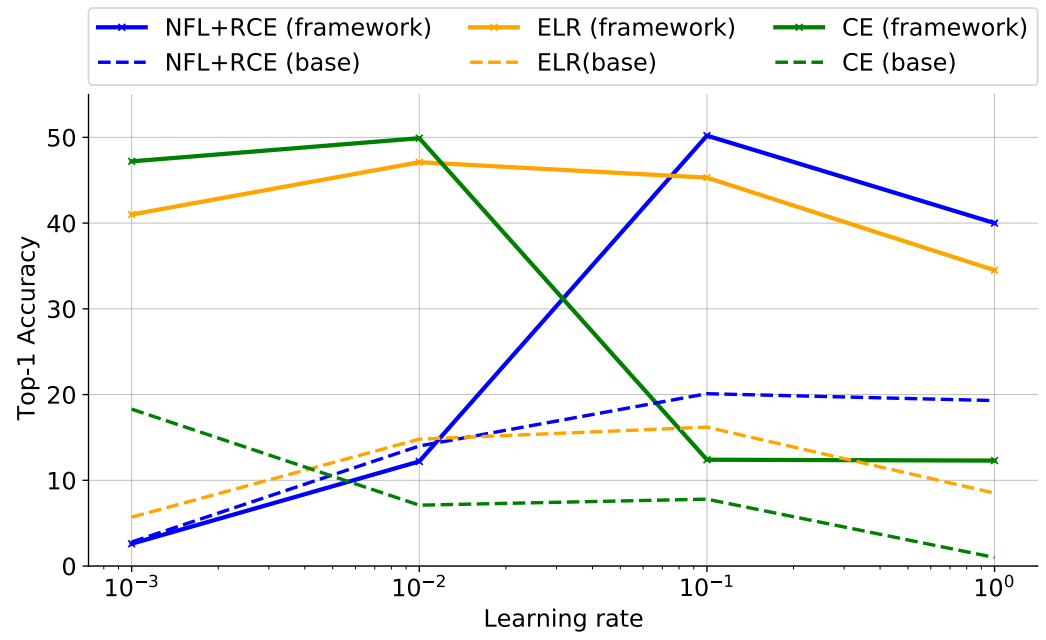
Table A3. Top-1 accuracy on CIFAR100 with 80% noise. Two different contrastive learning frameworks are evaluated for the pre-training: SimCLR and Moco. The third column gives the accuracy for a classifier with a smaller learning rate.

	SimCLR	Moco	Moco-Fine Tune
CE	12.4	12.0	49.0
ELR	45.3	38.8	42.3
NFL+RCE	50.2	26.3	47.0

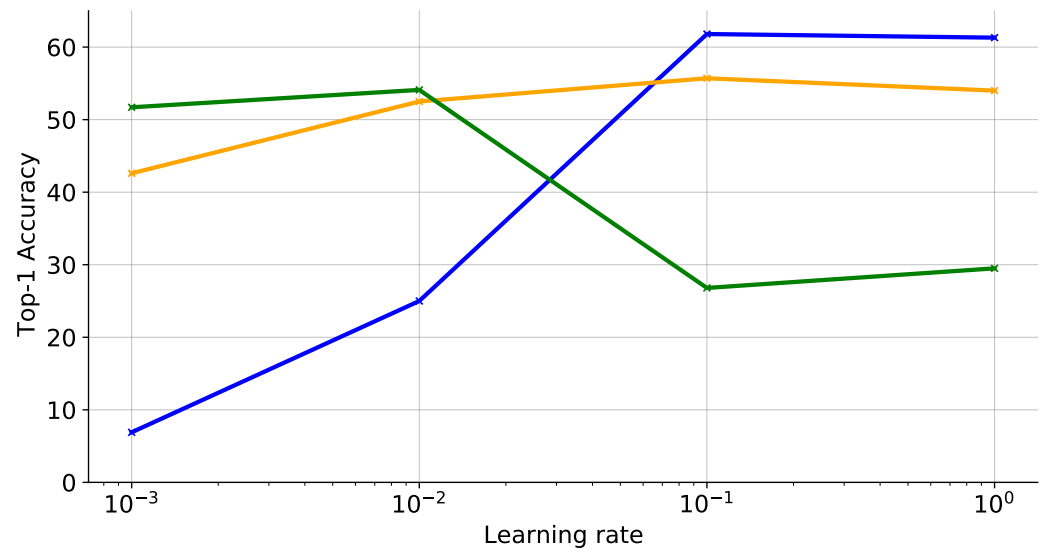
Even if pre-training the encoder increases the accuracy for both contrastive methods, the two approaches do not have the same behavior. In particular, the best parameters for the classifier optimizer seem to be different. This raises several questions about the difference between the two representations and what properties of these representations improve the robustness of the classifier.

Appendix C.2. Sensitivity to the Learning Rate

We perform a hyperparameter search on the CIFAR100 datasets. The learning rate is chosen in $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$. Results are presented in Figure A1. The configuration with 80% noise is clearly the most sensitive case, particularly for the NFL+RCE loss and the CE. The ELR method is quiet robust over the investigated range.

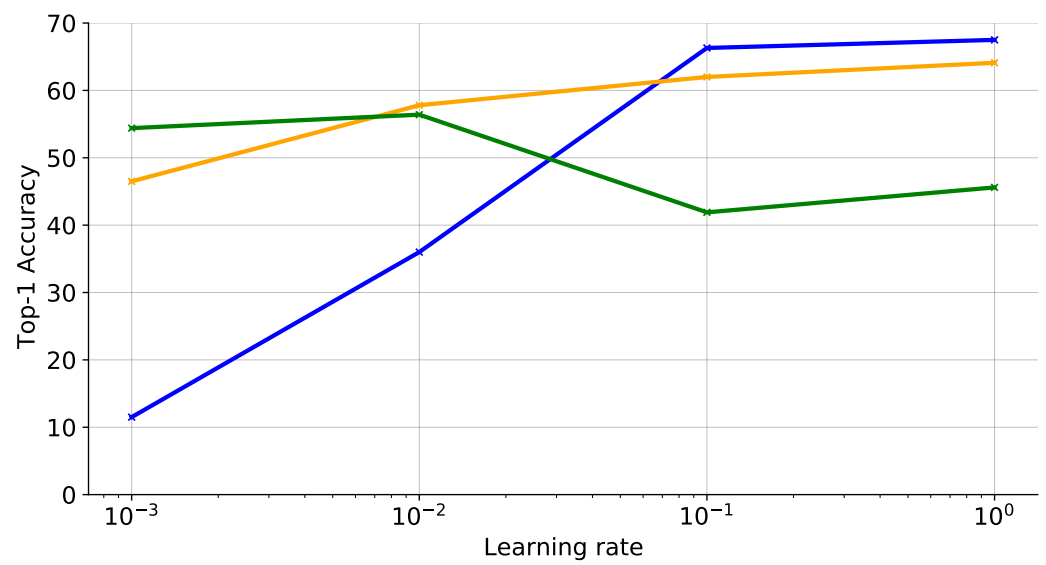


(a) CIFAR100 with 80% noise.

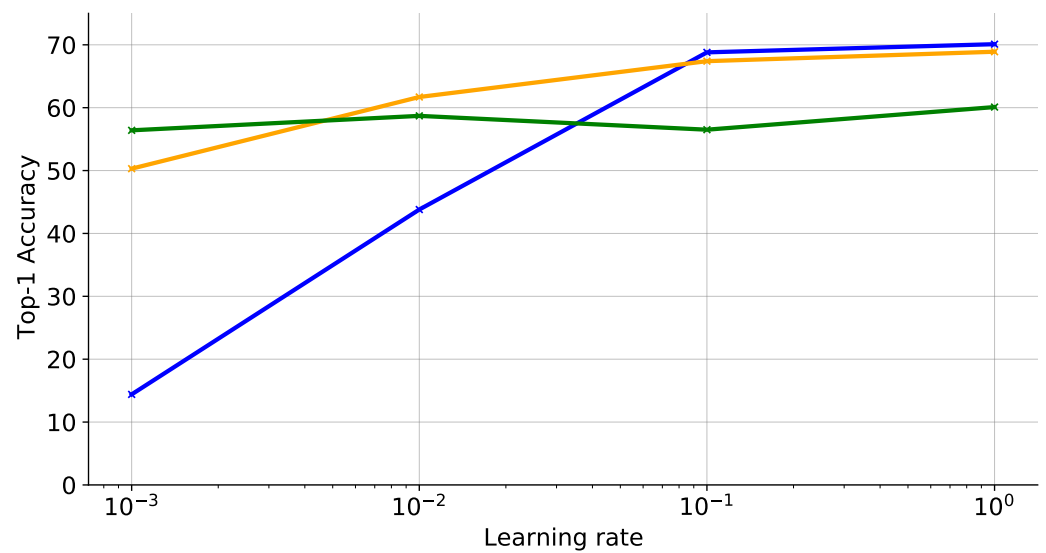


(b) CIFAR100 with 60% noise.

Figure A1. Cont.



(c) CIFAR100 with 40% noise.



(d) CIFAR100 with 20% noise.

Figure A1. Hyperparameter sensitivity for CIFAR100.

Appendix C.3. Impact of the Classifier Architecture

The impact of the 2 classifier architectures is detailed in Table A4. The multi-layer architecture performs better on datasets contaminated with a significant amount of asymmetric noise.

Table A4. Results on both CIFAR10 and CIFAR100 using symmetric noise (0.2–0.8) and asymmetric noise (0.2–0.4). We compare a single linear layer (L) to multiple layers (M) final classification head, for three losses: CE, ELR, and NFL+RCE.

Type	η	Loss	CIFAR10		CIFAR100	
			L	M	L	M
Sym	0.2	ce	91.7	87.7	58.6	56.5
		elr	92.9	93.0	66.4	67.4
		nfl_rce	93.2	92.7	69.7	68.8
	0.4	ce	90.6	78.0	44.2	41.9
		elr	92.1	92.0	60.8	62.0
		nfl_rce	92.1	91.4	67.0	66.3
	0.6	ce	88.1	59.2	28.9	26.8
		elr	89.7	90.4	54.0	55.7
		nfl_rce	90.2	88.1	63.7	61.8
	0.8	ce	72.6	27.3	14.1	12.4
		elr	82.0	84.8	41.6	45.3
		nfl_rce	78.9	59.9	54.2	50.2
Asym	0.2	ce	91.6	87.9	60.1	57.8
		elr	92.7	92.4	69.3	70.2
		nfl_rce	92.5	91.5	69.1	68.4
	0.3	ce	90.2	83.9	52.3	50.4
		elr	90.6	91.7	68.5	69.3
		nfl_rce	91.2	89.9	68.0	63.5
	0.4	ce	84.7	77.8	43.7	42.4
		elr	68.4	89.5	65.5	67.6
		nfl_rce	62.6	82.4	63.0	47.8

Appendix D. Dynamic Bootstrapping with Mixup

In addition to the presented fine-tuning phase, we also evaluated the performance of other techniques recently proposed for noisy label classification. The weights w computed by the sample selection phase can also be combined with a mixup data augmentation strategy [51]. A specific strategy for noisy labels, called dynamic bootstrapping with mixup [18], has been developed to help convergence under extreme label noise conditions. The convex combinations of sample pairs x_p (loss l_p) and x_q (loss l_q) is weighted by the probability w to belong to the clean dataset:

$$x = \frac{w_p}{w_p + w_q} x_p + \frac{w_q}{w_p + w_q} x_q. \quad (\text{A1})$$

$$l = \frac{w_p}{w_p + w_q} l_p + \frac{w_q}{w_p + w_q} l_q. \quad (\text{A2})$$

The associated CE is corrected according to the weights:

$$l_{ce} = - \sum_{k=1}^K (w_i q(k|x_i) + (1 - w_i) z_i) \log(p(k|x_i)), \quad (\text{A3})$$

where $z(k|x_i) = 1$ if $k = \operatorname{argmax} p(k|x_i)$ or zero for all the other cases. If the GMM probability is well estimated, combining one noisy sample with one clean sample leads to

a large weight for the clean sample and a small weight for the noisy sample. Clean–clean and noisy–noisy cases remain similar to a classical mixup with weights around 0.5.

The dynamic bootstrapping for ELR is derived by replacing the CE term by the corrected version:

$$l_{elr}(\theta) = l_{ceb}(\theta) + \frac{\lambda_{elr}}{N} \log \left(1 - \sum_{k=1}^K p(k|\mathbf{x}_i) \cdot t(k|\mathbf{x}_i) \right). \quad (\text{A4})$$

Regarding the robust loss function NFL+RCE, the two losses have to be modified:

$$l_{nfl} = w_i \frac{- \sum_{k=1}^K q(k|\mathbf{x}_i) (1 - p(k|\mathbf{x}_i))^\gamma \log(p(k|\mathbf{x}_i)) - \sum_{j=1}^K \sum_{k=1}^K q(y=j|\mathbf{x}_i) (1 - p(k|\mathbf{x}_i))^\gamma \log(p(k|\mathbf{x}_i))}{- \sum_{k=1}^K z(k|\mathbf{x}_i) (1 - p(k|\mathbf{x}_i))^\gamma \log(p(k|\mathbf{x}_i))} + (1 - w_i) \frac{- \sum_{k=1}^K z(k|\mathbf{x}_i) (1 - p(k|\mathbf{x}_i))^\gamma \log(p(k|\mathbf{x}_i)) - \sum_{j=1}^K \sum_{k=1}^K z(y=j|\mathbf{x}_i) (1 - p(k|\mathbf{x}_i))^\gamma \log(p(k|\mathbf{x}_i))}{- \sum_{j=1}^K \sum_{k=1}^K z(y=j|\mathbf{x}_i) (1 - p(k|\mathbf{x}_i))^\gamma \log(p(k|\mathbf{x}_i))} \quad (\text{A5})$$

where q is the one-hot encoding of the label (the zero value is fixed to a low value to avoid $\log(0)$).

$$l_{rce} = - \sum_{k=1}^K p(k|\mathbf{x}_i) \log(w_i \cdot q(k|\mathbf{x}_i) + (1 - w_i) z_i) \quad (\text{A6})$$

Appendix E. Classification Warmup

This section compares the classification accuracy of models trained with and without a warm-up phase after the representation learning. The warm-up phase consists of freezing the entire model except for the classification head. Figure A2 depicts the gain in performance brought by the warm-up phase. When using the default values, its inclusion is beneficial only for significant amounts of symmetric noise. Our experiments have been performed using only the recommended classifier learning rates, detailed in the experimental setup. Having different learning rates for the warm-up phase and the classification optimizing all weights (encoder and classifier) could have a different impact on the warmup phase.

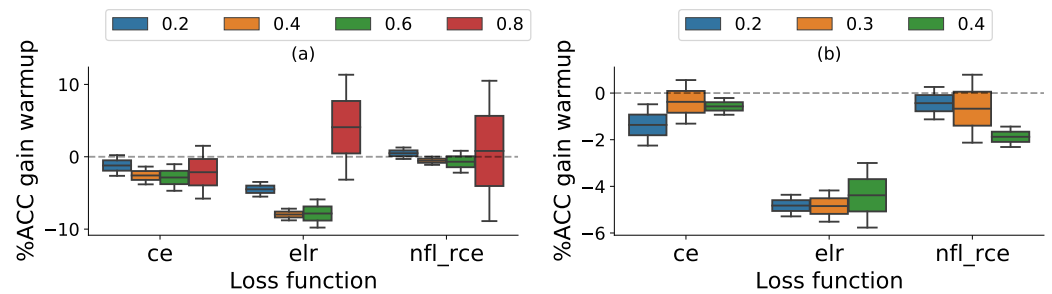


Figure A2. Gain in performance when using a supplementary classifier warm-up phase before training the entire model on CIFAR 100 with symmetric (panel a) and asymmetric noise (panel b).

Appendix F. Execution Time Analysis

In order to estimate our method's computational cost, we compared the execution time of both approaches, consisting of performing only the pre-training phase and the pre-training followed by fine-tuning with the execution time of performing only one supervised classification phase (i.e., the baseline). The number of times our methods were slower than the baseline is depicted in Table A5. We provided similar metrics for the

methods making available this information (i.e., Taks, Co-teaching+, JoCoR). As expected, the pre-training doubles the execution time of the baseline as, in addition to training the classifier, a contrastive learning phase has to be performed beforehand. The entire framework introduces a computational cost 3 to 4.5 times higher. However, all methods leveraging pre-trained models (using, for instance, supervised pre-training) also hide a similar computational cost.

Table A5. Comparison of execution time results reported as a factor with respect to the training time of the baseline, representing the supervised training of the model with the CE loss. The abbreviation Ours (Pre-t) indicates the pre-training phase while Ours (Fine-tune) indicates the pre-training phase followed by fine-tuning.

	C10 80% S	C10 40% A	C100 80% S	C100 40% A
Ours (Pre-t)	2.36	2.53	2.40	2.32
Ours (Fine-tune)	3.42	3.63	4.31	4.36
Taks	0.53	1.04	0.52	0.98
Co-teach+	2.00	2.00	2.00	2.01
JoCoR	1.73	1.74	1.72	1.74

Appendix G. An Attempt to Prevent Overfitting with Early Stopping

Overfitting is the common weakness of all studied models. Several strategies understanding and preventing overfitting have been explored: (i) analyzing the model behavior on a validation set, (ii) identifying the start of the memorization phase using Training Stop Point [48], and (iii) characterizing changes in the model using Centered Kernel Alignment [49]. A clean validation set is generally used to find the best moment for early stopping and to estimate the hyperparameter sets. However, we assume that clean validation samples are not available. Therefore, the methods must be robust to overfitting and to a wide range of hyperparameter values.

As typical noisy label settings lack a clean reference set, we contrasted the behavior of the model on a corrupted validation set with that on a clean test set, where overfitting can be easily identified. Train/validation sets have been generated using 5 cross validation folds. In the figure below, panel (a) depicts the evolution of accuracy scores on the corrupted train/validation sets as well as on the test set. After the first 50 epochs, the model starts overfitting as the test accuracy drops by 10% (Figure A3 panel a). The accuracy on the corrupted train continues to increase as the model memorizes the input labels. However, on the corrupted validation set a plateau followed by a loss of performance is indicative of the same phenomena, but without being always aligned with the overfitting phase observed on the test-set. The memorization phenomena of the train-set labels incapacitates the model to generalize on the corrupted validation set and explains the significant difference in scores between the train and validation accuracies.

A second perspective on the analysis of overfitting explores the stability of the network's predictions on the validation set. Panel (b) depicts the number of samples predicted in different classes across consecutive epochs. As the model starts overfitting, the prediction stability also increases. After 200 epochs, only 500 from 10,000 samples on the validation set change class from one epoch to another. As expected, the network stability is correlated with model overfitting on severe label noise.

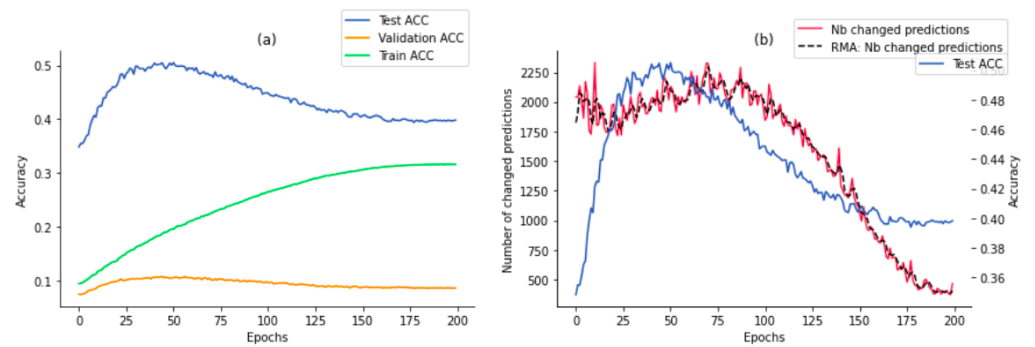


Figure A3. Evolution of accuracy across train/validation/test sets. (a) Prediction stability on the validation set computed as the number of samples changing class across consecutive epochs. We compared the stability of predictions (in red) with the accuracy of the clean test set (in blue). (b) The rolling mean average of the number of predictions has been depicted in black. The experiments have been performed on CIFAR 100, with 80% symmetric noise during the first classification phase and used NFL+RCE loss. In this plot, only the test set has correct labels. The accuracy on the corrupted validation set reflects the noise level while on the corrupted train set the increase in accuracy corresponds to overfitting (memorization of incorrect labels).

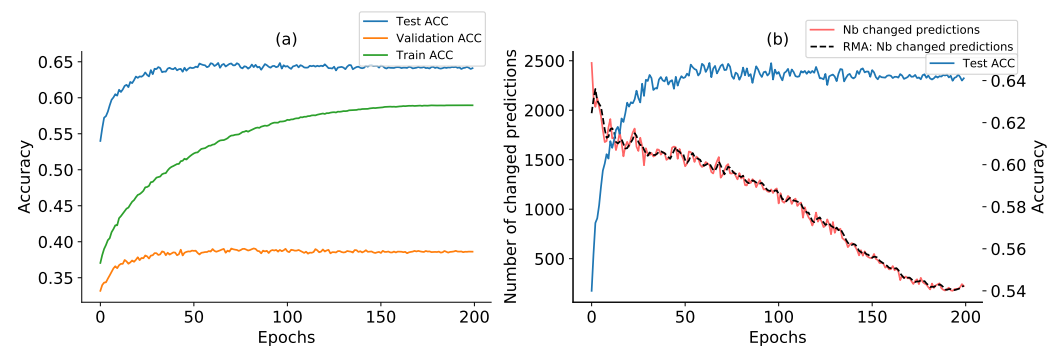


Figure A4. Evolution of accuracy across train/validation/test sets (a) Prediction stability on the validation set computed as the number of samples changing class across consecutive epochs. We compared the stability of predictions (in red) with the accuracy of the clean test set (in blue). (b) The rolling mean average of the number of predictions has been depicted in black. The experiments have been performed on CIFAR 100, with 40% asymmetric noise during the first classification phase and used NFL+RCE loss. In this plot, only the test set uses correct labels. The accuracy on the corrupted validation set reflects the level of label noise in the data, while on the corrupted train set the increase in accuracy corresponds to overfitting (memorization of incorrect labels).

Several recent contributions studied the overfitting phenomena of neural networks in an attempt to identify an early stopping point corresponding to the maximum obtainable test accuracy. Traditional approaches leverage a clean test set which is often unavailable when confronted with noisy labeled data. Kamabattula et al. [48] proposed to find a Training Stop Point (TSP), a heuristic analyzing the rate of change in the training accuracy and correlated its transition towards the memorization phase with a transition towards a smoother (smaller variance) regime, as depicted below. Our experimental results showed that the theoretical conditions to identify the early stopping point are not always met as suggested by TSP. Figure A5 indicates that the overfitting phase, starting after the first 5 epochs, does not change the variance of the train loss.

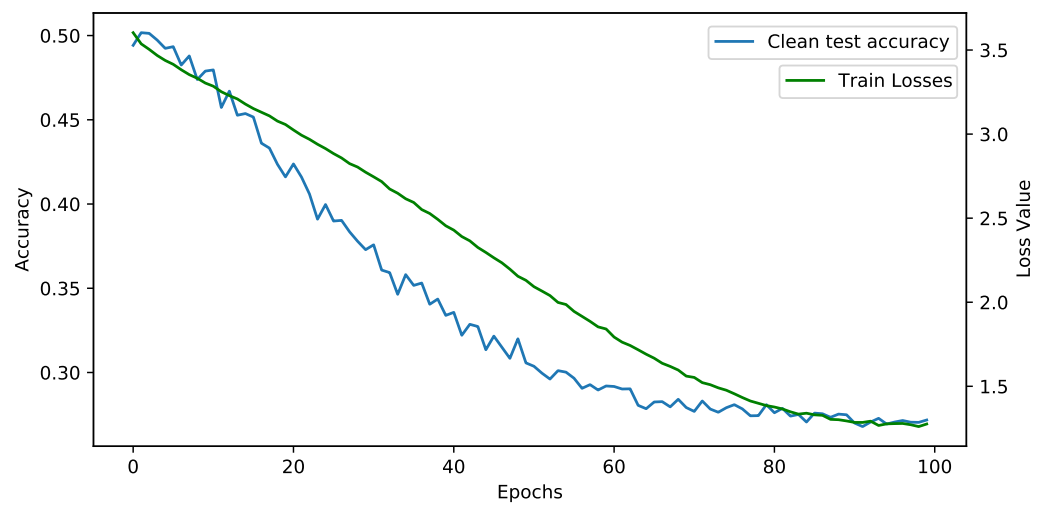


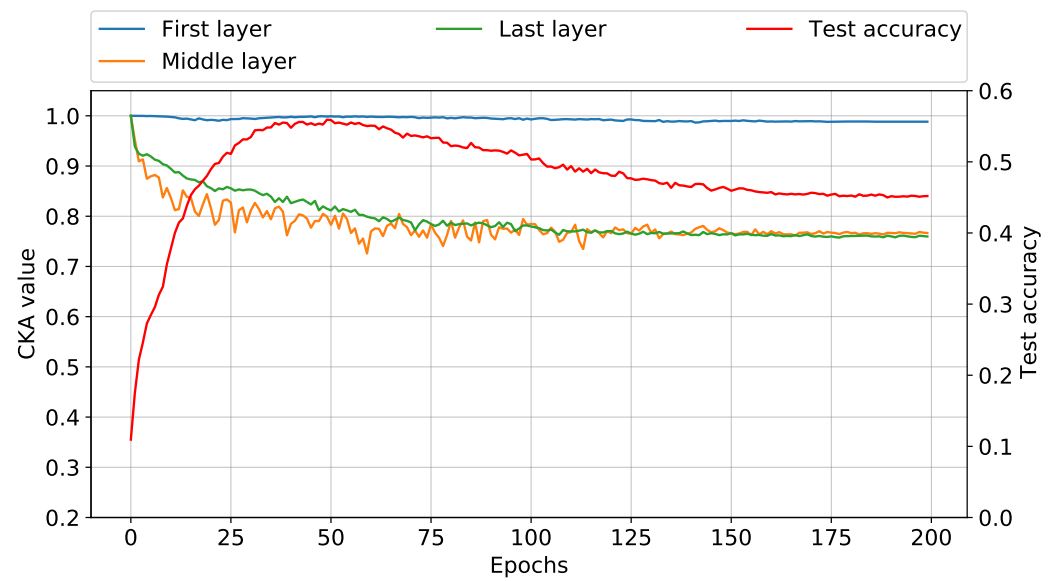
Figure A5. Evolution of train loss and test accuracy on CIFAR, 60% symmetric noise. The theoretical conditions of higher variance on the train loss, associated with the start of the memorization phase, as suggested by TSP, are not fulfilled.

Centered Kernel Alignment (CKA) [49] provides a similarity index comparing representations between layers of different trained models. In particular, CKA shows interesting properties as CKA can consistently identify correspondences between layers trained from different initializations.

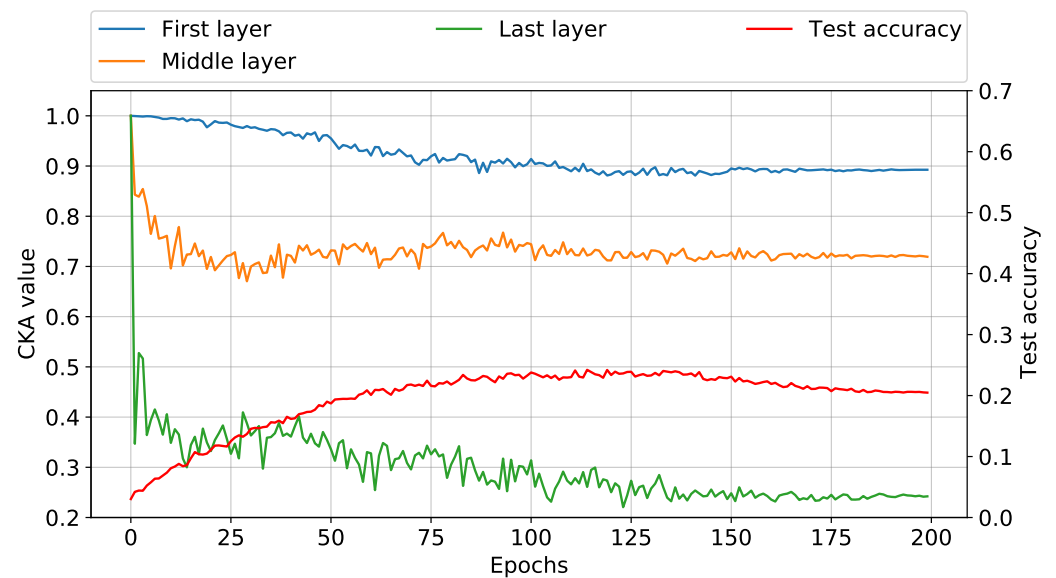
The objective is two-fold: (i) observing if a specific behavior can be identified for the overfitting and (ii) comparing the CKA values with and without contrastive pre-training. The CKA index is computed at three different locations in the network: the input layer, the middle of the network, and the final layer. Figure A6 shows the CKA similarity computed between the initialization/pre-trained model and the same layer at different epochs during the training process. It is interesting to note that the first layer of the pre-trained model remains very similar to the same layer computed by contrastive learning. Such behavior was expected in order to improve the robustness against noisy labels. Indeed, if contrastive learning can extract good representations for semi-supervised or transfer learning, being close to such representations can also help to avoid learning noisy labels. As expected, all layers of the model trained from a random initialization vary much more during the training.

The training phase of the pre-trained model reaches its maximum accuracy around 50 epochs but the CKA values of the middle and last layers continue to drop until 130 epochs. On the other hand, the CKA values of the initialized model remain stable after 150 epochs when the test accuracy reaches almost its maximum value. At first glance, the CKA behavior cannot be related to overfitting.

None of the studied approaches provides a solution preventing overfitting across all our experiments and this problem remains an open question.



(a) CKA from a pre-trained encoder with contrastive learning.



(b) CKA from a random initialization.

Figure A6. CKA similarity for a model trained with the NFL+RCE loss function on CIFAR100 with 80% noise.

References

1. Mahajan, D.; Girshick, R.; Ramanathan, V.; He, K.; Paluri, M.; Li, Y.; Barambe, A.; van der Maaten, L. Exploring the limits of weakly supervised pretraining. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 181–196.
2. Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv* **2016**, arXiv:1611.03530.
3. Patrini, G.; Rozza, A.; Krishna Menon, A.; Nock, R.; Qu, L. Making deep neural networks robust to label noise: A loss correction approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1944–1952.
4. Goldberger, J.; Ben-Reuven, E. Training Deep Neural-Networks Using a Noise Adaptation Layer. ICLR. 2017. Available online: <https://openreview.net/forum?id=H12GRgcxg> (accessed on 15 June 2020).

5. Xia, X.; Liu, T.; Wang, N.; Han, B.; Gong, C.; Niu, G.; Sugiyama, M. Are anchor points really indispensable in label-noise learning? In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 6838–6849.
6. Hendrycks, D.; Mazeika, M.; Wilson, D.; Gimpel, K. Using trusted data to train deep networks on labels corrupted by severe noise. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 10456–10465.
7. 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 Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 2304–2313.
8. 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. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; pp. 8527–8537.
9. Li, J.; Socher, R.; Hoi, S.C. DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In Proceedings of the International Conference on Learning Representations, Virtual Event, 26 April–1 May 2020.
10. Zhang, Z.; Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 8778–8788.
11. Wang, Y.; Ma, X.; Chen, Z.; Luo, Y.; Yi, J.; Bailey, J. Symmetric cross entropy for robust learning with noisy labels. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 322–330.
12. Ma, X.; Huang, H.; Wang, Y.; Romano, S.; Erfani, S.; Bailey, J. Normalized loss functions for deep learning with noisy labels. In Proceedings of the 37th International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 6543–6553.
13. Liu, S.; Niles-Weed, J.; Razavian, N.; Fernandez-Granda, C. Early-Learning Regularization Prevents Memorization of Noisy Labels. In Proceedings of the Advances in Neural Information Processing Systems, Online, 6–12 December 2020; Volume 33.
14. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv* **2020**, arXiv:2002.05709.
15. 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, Virtual Event, 14–19 June 2020; pp. 9729–9738.
16. Song, H.; Kim, M.; Park, D.; Lee, J.G. Learning from noisy labels with deep neural networks: A survey. *arXiv* **2020**, arXiv:2007.08199.
17. Le-Khac, P.H.; Healy, G.; Smeaton, A.F. Contrastive representation learning: A framework and review. *IEEE Access* **2020**, *8*, 193907–193934. [[CrossRef](#)]
18. Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N.; McGuinness, K. Unsupervised label noise modeling and loss correction. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 312–321.
19. Song, H.; Kim, M.; Lee, J.G. Selfie: Refurbishing unclean samples for robust deep learning. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 5907–5915.
20. Arpit, D.; Jastrzebski, S.; Ballas, N.; Krueger, D.; Bengio, E.; Kanwal, M.S.; Maharaj, T.; Fischer, A.; Courville, A.; Bengio, Y.; et al. A closer look at memorization in deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 233–242.
21. Nguyen, D.T.; Mummadi, C.K.; Ngo, T.P.N.; Nguyen, T.H.P.; Beggel, L.; Brox, T. SELF: Learning to Filter Noisy Labels with Self-Ensembling. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
22. Wang, Z.; Jiang, J.; Han, B.; Feng, L.; An, B.; Niu, G.; Long, G. SemiNLL: A Framework of Noisy-Label Learning by Semi-Supervised Learning. *arXiv* **2020**, arXiv:2012.00925.
23. Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; Raffel, C.A. Mixmatch: A holistic approach to semi-supervised learning. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; pp. 5049–5059.
24. Ortego, D.; Arazo, E.; Albert, P.; O'Connor, N.E.; McGuinness, K. Multi-Objective Interpolation Training for Robustness to Label Noise. *arXiv* **2020**, arXiv:2012.04462.
25. Zhang, H.; Yao, Q. Decoupling Representation and Classifier for Noisy Label Learning. *arXiv* **2020**, arXiv:2011.08145.
26. Li, J.; Xiong, C.; Hoi, S.C. MoPro: Webly Supervised Learning with Momentum Prototypes. *arXiv* **2020**, arXiv:2009.07995.
27. Henaff, O. Data-efficient image recognition with contrastive predictive coding. In Proceedings of the International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 4182–4192.
28. Misra, I.; Maaten, L.V.D. Self-supervised learning of pretext-invariant representations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual Event, 14–19 June 2020; pp. 6707–6717.
29. Kalantidis, Y.; Sariyildiz, M.B.; Pion, N.; Weinzaepfel, P.; Larlus, D. Hard negative mixing for contrastive learning. *arXiv* **2020**, arXiv:2010.01028.
30. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. Bootstrap Your Own Latent—A New Approach to Self-Supervised Learning. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 21271–21284.
31. Chen, X.; Fan, H.; Girshick, R.; He, K. Improved baselines with momentum contrastive learning. *arXiv* **2020**, arXiv:2003.04297.

32. Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; Joulin, A. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In Proceedings of the Thirty-Fourth Conference on Neural Information Processing Systems (NeurIPS), Online, 6–12 December 2020.
33. Ghosh, A.; Kumar, H.; Sastry, P. Robust loss functions under label noise for deep neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31.
34. Ouahabi, A.; Taleb-Ahmed, A. Deep learning for real-time semantic segmentation: Application in ultrasound imaging. *Pattern Recognit. Lett.* **2021**, *144*, 27–34. [[CrossRef](#)]
35. Zadeh, S.G.; Schmid, M. Bias in cross-entropy-based training of deep survival networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**. [[CrossRef](#)]
36. Falcon, W.; Cho, K. A framework for contrastive self-supervised learning and designing a new approach. *arXiv* **2020**, arXiv:2009.00104.
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26–30 June 2016; pp. 770–778.
38. Wu, Z.; Xiong, Y.; Yu, S.X.; Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 3733–3742.
39. Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised Contrastive Learning. In Proceedings of the Advances in Neural Information Processing Systems, Online, 6–12 December 2020; Volume 33.
40. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. Master’s Thesis, University of Toronto, Toronto, ON, Canada, 2009.
41. Li, W.; Wang, L.; Li, W.; Agustsson, E.; Van Gool, L. Webvision database: Visual learning and understanding from web data. *arXiv* **2017**, arXiv:1708.02862.
42. Xiao, T.; Xia, T.; Yang, Y.; Huang, C.; Wang, X. Learning from massive noisy labeled data for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 8–10 June 2015; pp. 2691–2699.
43. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d’Alch’e-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.
44. Song, H.; Mitsuo, N.; Uchida, S.; Suehiro, D. No Regret Sample Selection with Noisy Labels. *arXiv* **2020**, arXiv:2003.03179.
45. Yu, X.; Han, B.; Yao, J.; Niu, G.; Tsang, I.; Sugiyama, M. How does disagreement help generalization against label corruption? In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 7164–7173.
46. Wei, H.; Feng, L.; Chen, X.; An, B. Combating noisy labels by agreement: A joint training method with co-regularization. *arXiv* **2020**, arXiv:2003.02752.
47. Bianco, S.; Cadene, R.; Celona, L.; Napolitano, P. Benchmark analysis of representative deep neural network architectures. *IEEE Access* **2018**, *6*, 64270–64277. [[CrossRef](#)]
48. Kamabattula, S.R.; Devarajan, V.; Namazi, B.; Sankaranarayanan, G. Identifying Training Stop Point with Noisy Labeled Data. *arXiv* **2020**, arXiv:2012.13435.
49. Kornblith, S.; Norouzi, M.; Lee, H.; Hinton, G. Similarity of neural network representations revisited. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 3519–3529.
50. Mitrovic, J.; McWilliams, B.; Rey, M. Less can be more in contrastive learning. In Proceedings of the “I Can’t Believe It’s Not Better!” NeurIPS 2020 Workshop, Virtual Event, 6–14 December 2020.
51. Zhang, H.; Cissé, M.; Dauphin, Y.; Lopez-Paz, D. mixup: Beyond Empirical Risk Minimization. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.