Label Noise Resiliency with Self-supervised Representations

Zahra Vaseqi[*]
McGill University
zahra@cim.mcgill.ca

Ibtihel Amara[*]
McGill University
ibtihel.amara@mail.mcgill.ca

Samrudhdhi Rangrej[*]
McGill University
samrudhdhi.rangrej@mail.mcgill.ca

Abstract

Training deep networks with noisy labels leads to inaccurate modeling and poor generalization capability due to overfitting the noise. One technique to overcome this challenge is to pretrain a feature extractor backbone without labels using self-supervised learning (SSL) and training only a linear classifier on the noisy labels. Another technique to pretrain the feature extractor is to perform supervised learning on an alternate clean in-domain dataset. Here, we assess the robustness of representations learned using various SSL methods towards the noise in the label set and compare it with the noise robustness achieved using supervised training on an alternate dataset. We present a small-scale and yet detailed study of five SSL methods on CIFAR-10 dataset for symmetric and asymmetric noise. We find that SimCLR and MoCo respectively achieve the most and the least robustness; however, they both outperform the supervised learning in terms of noise resiliency.

1 Introduction

Recent success of deep learning can largely be attributed to advances in supervised learning on large-scale labelled datasets. However, collecting high quality labels is an expensive endeavour; especially, when expert knowledge is required for annotation. On the other hand, there exist many alternatives such as crowdsourcing which facilitates affordable annotation, but with certain amount of noise in the labelling processes. Label-noise is inevitable either due to accidental mistakes or caused by the inherently incomprehensible ambiguities in the data.

Supervised training with label-noise leads to inaccurate representation learning as the model overfits to such noise [1]. There are many approaches to circumvent this issue. For example, recent methods propose verifying labels on few images and estimating label confusion [2], loss correction technique [3], reweighting examples [4], true label estimation [5], noise adaptation layer [6], curriculum learning [7], etc. In this paper we focus on learning a feature extractor, or backbone, without using the noisy labels and then training a linear classifier for the pretrained and frozen backbone using the noisy labels. As only the last linear layer is trained with the noisy labels, this technique reduces overfitting.

Specifically, we analyze two approaches where the backbone is trained using (i) contrastive learning with unlabeled data and (ii) supervised learning with the alternate clean in-domain dataset. Unlike the first approach, the second approach requires a clean label set from the in-domain dataset. Note that the in-domain dataset is different from the noisy dataset but includes images from the same domain.

[*]Equal contribution.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

In essence, the second approach is akin to transfer learning. We assess these approaches for two types of label noise, namely, symmetric and asymmetric. The symmetric noise corrupts the labels in a purely random manner using a uniform distribution. Meanwhile, asymmetric noise simulates naturally occurring label noise where two classes with similar features (ex. cats and dogs) may be confused and thereby mislabeled.

The main contributions of this paper are as follows. (i) We study five contrastive learning methods in terms of their robustness to label-noise. We assess robustness to two types of noise, i.e. symmetric and asymmetric, at ten different levels ranging from 0.0 to 0.9 on CIFAR-10 dataset. (ii) We characterize the robustness gap across the two highest-performing SSL models based on the learnt feature representations. (iii) We compare the noise robustness behavior of self-supervised methods with the ones achieved using supervised transfer learning and find the former to be more robust than the latter.

2 Experiments and Analysis

First, we pretrain the five self-supervised models, namely, SimCLR [8], MoCo [9], BYOL [10], SimSiam [11], and SwaV [12]. Then we train a linear classifier on top of the pretrained models using noisy labels. We compare the models in terms of their noise-resiliency against symmetric and asymmetric noise. Next, we compare the most robust model with a supervised model of equal capacity trained on an equivalent dataset. In the next section, we will elaborate our experimental setup and the datasets we used for training.

2.1 Dataset Preparation for Noisy Labels.

We perform experiments on CIFAR-10 [13] and artificially inject two types of label noise. A noise rate parameter η determines the proportion of the training examples that are corrupted. To avoid class imbalance issues, we corrupt each class with the same noise ratio. We set the noise rate η to 10 values between 0.0 and 1.0 with a uniform intervals, $\eta \in \{0.0, 0.1, 0.2, \ldots, 0.9\}$. The clean dataset is shown with $\eta = 0.0$. Given a dataset with $y \in \{1, \ldots, K\}$ classes, we define a label transition matrix T where t_{ij} is the probability of having label $y = i$ corrupted by $y = j$ where $i \neq j$ and $i, j \in \{1, \ldots, K\}$. For the case with the *symmetric noise*, is a purely random noise, where the label y is corrupted with a uniform probability $t_{ij} = \frac{1}{K-1}$ to one of the other $K - 1$ classes –which excludes the true class label. The *asymmetric noise* is a class-dependent noise often caused by confusing a label for another one from a similar class. In this setting, the probabilities for label transitions among some classes are higher than others; e.g., class "dog" may be more likely to be mistaken for "cat", rather than "car". To simulate this type of naturally occurring noise, we corrupt the label for a given class using the label from its subsequent class, i.e. $y = 1 \rightarrow 2 \rightarrow \ldots \rightarrow K \rightarrow 1$. In order to remove the effect of the randomly generated noisy training set on our models, we use a fixed random seed when generating our noisy training sets to ensure all models that are being compared against each other are presented with the same noisy dataset.

2.2 Resiliency to Label Noise

In this experiment, we examine the behaviour of self-supervised methods towards noisy data labels on a classification downstream task. In particular, we analyze the effect of the different types of label noise on their performance.

Setup. We train all models using `lightly`[14], a computer vision framework for self-supervised learning. Specifically, we use `cifar10_benchmark.py`[2] with the default hyperparameter setting provided by `lightly`. We train all models with batch size of 512 for 200 epochs on CIFAR-10 train set. All methods are trained with ResNet-18 backbone[15].

k-NN accuracy on clean CIFAR-10 test set upon the completion of pretraining is as follows: SimCLR-83.52%, SwAV-70.57%, MoCo-84.29%, SimSiam-80.80%, BYOL-71.05%. We consider the last checkpoint for each method and remove projection/prediction head(s). We use two fully connected layers for the projection head: MoCo, SimCLR, and SwAV use ReLU nonlinearity on the first projection layer, while SimSiam, BarlowTwins, and BYOL apply BatchNorm followed by ReLU activation. Then, we compose the pretrained ResNet-18 with a linear classifier initialized with random

[2] `https://docs.lightly.ai/_downloads/b99fe89a7fc2b4740cb9f1e34d3229ad/cifar10_benchmark.py`

Noise	Symmetric Noise					Asymmetric Noise				
(%)	MoCo	SimCLR	SwAV	SimSiam	BYOL	MoCo	SimCLR	SwAV	SimSiam	BYOL
00	83.57	81.92	66.71	80.46	67.29	83.57	81.92	66.71	80.46	67.29
10	82.15	81.24	65.11	79.53	65.67	83.43	81.84	66.52	80.34	66.98
20	80.55	80.37	63.57	78.32	64.56	82.95	81.62	66.13	80.07	66.70
30	78.62	79.13	61.36	76.50	62.65	81.90	81.15	64.76	79.27	65.67
40	75.82	77.05	60.07	73.90	61.11	80.05	80.06	63.36	78.15	64.14
50	73.26	75.24	57.82	72.28	59.15	76.77	77.96	61.37	74.82	62.29
60	70.24	71.56	55.25	68.49	57.00	73.12	73.91	59.11	72.13	59.20
70	67.28	70.06	54.33	65.12	55.12	69.50	70.24	55.40	67.44	56.28
80	62.65	65.71	51.16	62.42	52.89	62.87	65.05	51.97	62.26	51.74
90	57.62	60.38	47.41	57.45	47.79	56.38	57.40	46.82	55.51	47.09

Table 1: Linear Classification test accuracy(%) on noisy CIFAR-10 using Resnet-18 backbone.

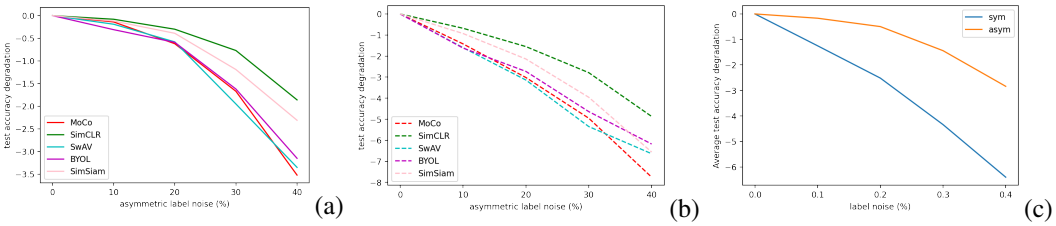


Figure 1: Average test accuracy (%) degradation in pretrained SSL models using (a) symmetric and (b) asymmetric noise. (c) Average accuracy degradation across SSL models for two noise-types.

weights. We normalize features from ResNet-18 before passing them to the linear classifier[3]. We train the linear classifier while keeping the weights for ResNet-18 backbone frozen. Training of a linear classifier is performed using corrupted label set, as described in Section 2.1, and the final model is tested on clean test set.

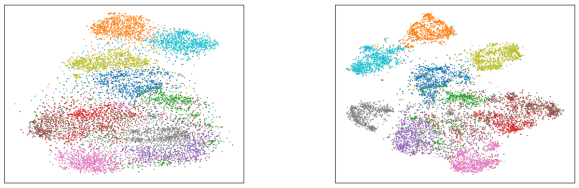
Results. MoCo has the highest top 1% accuracy (84.29%) overall and SimCLR coming in second highest (83.52%) using a KNN classifier with regards to clean target labels (no target noise). We also find the same performance trend when we use a linear classifier in Table 1 at 0.0 noise rate.

Figure 1 (c), we notice that in total, symmetric noise has stronger influence on the models' performance than asymmetric noise. Unlike symmetric noise, asymmetric noise is more structured (as detailed in Section 2.1) and hence easy to resist.

Figure 1 (a) and (b) show the level of degradation of the test accuracy of the self-supervised pretrained models using asymmetric noise and symmetric noise, respectively. We observe that SimCLR is less sensitive overall to asymmetric and symmetric noise when compared to the other models. Its performance, in terms of accuracy, decreased by 1.8% from 0% to 40% asymmetric noise rate and by 4.8% for symmetric noise. On the other hand, MoCo's performance dropped by 3.5%, which is almost twice the drop of SimCLR, from 0% to 40% asymmetric noise rate and 8% symmetric noise (worst performance overall methods). Other methods, i.e., SwAV, SimSiam and BYOL fall between MoCo and SimCLR. The question that we could ask here is why did MoCo's performance drop: from being the best performing model overall on noise-free data to being the worst performing model? To investigate the root cause of this decline in performance, we provided in Figure 2 (a) and Figure 2 (b) the T-SNE visualization for each of MoCo and and SimCLR, respectively, on the test set image features color coded according to the true (no noise) class labels. We observe that the classes visualizations of the embedding distribution for SimCLR tends to generate tighter distributions and be more tolerant to slight variations in noise levels. MoCo's classes embedding distribution, on the other hand, shows a much more uniform distribution. In this scenario, a slight variation in noise levels can cause the class labels to intertwine causing a substantial decrease in performance.

We speculate that MoCo generates relatively sparse clusters, compared to SimCLR, since its temperature in the InfoNCE loss is set lower. MoCo is trained with a temperature of 0.1 and SimCLR with 0.5. Based on this hypothesis, we conceive and perform an ablation on the temperature hyperparameter for MoCo. See results in Figure 3. We observe that increasing the value of the temperature parameter up to a certain value increases the robustness. MoCo achieves the highest noise resiliency at a temperature of 0.5, similar to SimCLR.

[3] We follow 'Tutorial 2: Train MoCo on CIFAR-10' (`https://docs.lightly.ai/tutorials/package/tutorial_moco_memory_bank.html`)for training a linear classifier on top of pretrained backbone.



(a) (b)

Figure 2: T-SNE [16] visualization of CIFAR-10 test set image features, color coded according to the true class labels (a) MoCo (b) SimCLR.

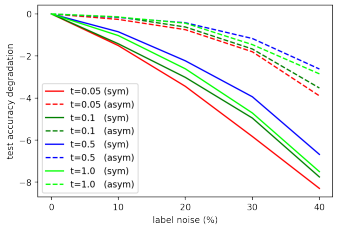


Figure 3: Test accuracy (%) degradation of MoCo pretrained model with different temperature values.

Noise	Symmetric Noise			Asymmetric Noise		
(%)	TL	MoCo	SimCLR	TL	MoCo	SimCLR
0	85.04	75.96	76.83	85.04	75.96	76.83
10	74.00	74.09	75.71	83.75	75.93	76.51
20	64.54	71.77	73.85	80.67	75.40	76.32
30	59.09	69.53	71.68	75.40	74.05	75.15
40	54.49	66.79	69.59	69.02	71.83	73.58
50	48.74	64.46	66.88	65.57	69.25	71.14
60	43.75	61.47	64.48	58.35	65.09	68.39
70	38.78	58.14	61.54	54.03	60.92	63.63
80	36.14	54.76	57.77	48.96	55.89	59.84
90	32.85	50.44	53.73	43.28	50.95	52.44

Table 2: Accuracy (%) of transfer learning (TL) and SSL methods on clean CIFAR-10 test set.

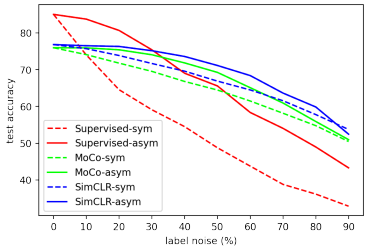


Figure 4: Test accuracy (%) of supervised pretraining compared against self-supervised pretraining

2.3 Transfer learning

In this section, we compare robustness of MoCo and SimCLR method against transfer learning using supervised pretraining.

Setup. We pretrain a backbone with a linear classifier on in-domain clean dataset using supervised learning. Next, we finetune the final linear classifier on noisy dataset. To control the unknown domain shift and to make the comparison between the SSL methods and the transfer learning method fair, we create the in-domain dataset as follows: we split the CIFAR-10 training set and split it randomly into two equal subsets, as described in Section 2.1. We use the first subset as an in-domain dataset. We inject label noise in the second subset and refer to it as noisy dataset.

We pretrain a ResNet-18 backbone on clean in-domain subset using SGD optimizer with 0.01 learning rate and 0.0001 weight decay for 100 epoch while using minibatches of size 128. Once trained, we freeze the backbone network and continue training the final linear classifier on noisy subset using SGD optimizer with 30.0 learning rate and 0.0001 weight decay for 100 epoch while using minibatch size of size 128. Similarly, we train MoCo and SimCLR following the same setup as discussed in Section 2.2. We initially train the models using only 50% of the training data–that is clean and in the case with self-supervised pretraining it is unlabeled. We use the remaining unseen 50% of the data for finetuning–the noise is added only to this second half of the data.

Results. Table 2 provides a summary of our results and Figure 4 depicts the performance degradation trend across the three models. The model with the supervised pretraining achieves the highest test accuracy; but it suffers a dramatic performance drop when finetuned on noisy data.

3 Discussion and Conclusion

In this paper, we studied five SSL methods in terms of their capability in learning representations that are robust to label noise. Overall, we find that the learnt representations are more robust to the asymmetric noise than the symmetric noise. Among the SSL methods, representations learnt using SimCLR and MoCo achieve the most and the least robustness. Furthermore, we find that the temperature in the InfoNCE loss plays an important role. We also compared SimCLR and MoCo against supervised pretraining on clean dataset. Although a model with supervised pretraining outperforms its self-supervised counterparts with a high margin at noise-free setup, it lacks robustness and suffers a dramatic performance drop when finetuned on noisy data. Our study establishes that one should prefer SSL pretraining over supervised pretraining in noisy data regimes and tune the temperature parameter of InfoNCE loss to achieve high noise resiliency.

References

[1] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021.

[2] Kuang-Huei Lee, Xiaodong He, Lei Zhang, and Linjun Yang. Cleannet: Transfer learning for scalable image classifier training with label noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5447–5456, 2018.

[3] Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1944–1952, 2017.

[4] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pages 4334–4343. PMLR, 2018.

[5] Daiki Tanaka, Daiki Ikami, Toshihiko Yamasaki, and Kiyoharu Aizawa. Joint optimization framework for learning with noisy labels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5552–5560, 2018.

[6] Jacob Goldberger and Ehud Ben-Reuven. Training deep neural-networks using a noise adaptation layer. In *ICLR*, 2017.

[7] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pages 2304–2313. PMLR, 2018.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 2020.

[9] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *CoRR*, abs/1911.05722, 2019.

[10] Pierre H. Richemond, Jean-Bastien Grill, Florent Altché, Corentin Tallec, Florian Strub, Andrew Brock, Samuel L. Smith, Soham De, Razvan Pascanu, Bilal Piot, and Michal Valko. BYOL works even without batch statistics. *CoRR*, abs/2010.10241, 2020.

[11] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

[12] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[14] Lightly - documentation. `https://docs.lightly.ai/`. Accessed: 2021-10-06.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[16] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.