Investigating Why Contrastive Learning Benefits Robustness against Label Noise

Anonymous Authors[1]

Abstract

Self-supervised contrastive learning has recently been shown to be very effective in preventing deep networks from overfitting noisy labels. Despite its empirical success, the theoretical understanding of the effect of contrastive learning on boosting robustness is very limited. In this work, we rigorously prove that learned the representation matrix has certain desirable properties in terms its SVD that benefit robustness against label noise. We further show that the low-rank structure of the Jacobian of deep networks pre-trained with contrastive learning allows them to achieve a superior performance initially, when fine-tuned on noisy labels. Finally, we demonstrate that the initial robustness provided by contrastive learning enables robust training methods to achieve state-of-the-art performance under extreme noise levels.

1. Introduction

Very recently, self-supervised contrastive learning has shown a lot of promise in boosting robustness of deep networks against noisy labels (Zheltonozhskii et al., 2022; Hendrycks et al., 2019; Ghosh & Lan, 2021). Despite its empirical success, the theoretical understanding of the effect of contrastive learning on improving robustness of deep networks against noisy labels is very limited.

In this work, we derive rigorous guarantees for the robustness of deep networks trained with contrastive learning against label noise. In particular, we study the effect of contrastive learning on robustness of deep networks trained on noisy labels perturbed with either Gaussian noise, or random label flipping. We prove that contrastive learning boosts robustness under extreme noise by learning a representation matrix that has: (i) a prominent singular value corresponding to each sub-class in the data, and very small remaining singular values; and (ii) a large alignment between the prominent singular vector and the clean labels of each sub-class. As a result, training a linear layer with representations obtained from contrastive learning quickly decreases the error over different sub-classes and allows the network to correctly classify wrongly-labeled examples.

We further show that deep networks pre-trained with contrastive learning can be fine-tuned on noisy labels to achieve a superior performance initially, before overfitting the noise. We attribute this to the initial low-rank structure of the Jacobian with a few large singular values associated to its prominent directions and insignificant singular values otherwise.

Finally, we conduct extensive experiments on noisy CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009) and the mini Webvision datasets (Li et al., 2017) to demonstrate that the initial robustness provided by contrastive learning can be further leveraged by robust methods to achieve state-of-the-art performance under extreme levels of noise.

2. Problem Formulation and Background

Suppose we have a dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$, where $(\boldsymbol{x}_i, \boldsymbol{y}_i)$ denotes the i-th sample with input $\boldsymbol{x}_i \in \mathbb{R}^d$ and its clean one-hot encoded label $\boldsymbol{y}_i \in \mathbb{R}^K$ corresponding to one of the K classes. For example, for a data point $\boldsymbol{x}_i$ from class $j \in [K]$, we have $\boldsymbol{y}_i = \boldsymbol{e}_j$ where $\mathbf{e}_j$ denotes the vector with a 1 in the jth coordinate and 0's elsewhere. We further assume that there are $\bar{K} \geq K$ sub-classes in the data. Sub-classes of a class share the same label, but are distinguishable from each other. We assume that for every data point $\boldsymbol{x}_i$, we only observe a noisy version of its label $\hat{\boldsymbol{y}}_i$. We denote by $\boldsymbol{Y}, \hat{\boldsymbol{Y}} \in \mathbb{R}^{n \times K}$ the matrices of all the one-hot encoded clean and noisy labels of the training data points.

2.1. Self-supervised Contrastive Learning

Self-supervised contrastive learning learns representations of different data points by maximizing agreement between differently augmented views of the same example via a contrastive loss in the latent space, as we discuss below.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the First Workshop of Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022. Do not distribute.

Augmentation graph. The augmentations can be used to construct the *population augmentation graph* (HaoChen et al., 2021), whose vertices are all the augmented data in the population distribution, and two vertices are connected with an edge if they are augmentations of the same natural example. Hence, ground-truth classes naturally form connected sub-graphs. Formally, let P be the distribution of all natural data points (raw inputs without augmentation). For a natural data point $\boldsymbol{x}^* \sim P$, let $\mathcal{A}(\cdot|\boldsymbol{x}^*)$ be the distribution of $\boldsymbol{x}^*$'s augmentations. For instance, when $\boldsymbol{x}^*$ represents an image, $A(.|\boldsymbol{x}^*)$ can be the distribution of common augmentations (Chen et al., 2020) including Gaussian blur, color distortion and random cropping. Then, for an augmented data point $\boldsymbol{x}$, $\mathcal{A}(\boldsymbol{x}|\boldsymbol{x}^*)$ is the probability of generating $\boldsymbol{x}$ from $\boldsymbol{x}^*$. The edge weights $w_{x_i x_j} = \mathbb{E}_{\boldsymbol{x}^* \sim P}[\mathcal{A}(\boldsymbol{x}_i|\boldsymbol{x}^*)\mathcal{A}(\boldsymbol{x}_j|\boldsymbol{x}^*)]$ can be interpreted as the marginal probability of generating $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ from a random natural data point.

Contrastive loss. The embeddings produced by contrastive learning can be viewed as a low-rank approximation of the normalized augmentation graph. Effectively, minimizing a loss that performs spectral decomposition on the population augmentation graph can be succinctly written as a contrastive learning objective $\mathfrak{C}(f)$ on neural network representations (HaoChen et al., 2021):

$$\mathfrak{C}(f) = -2\mathbb{E}_{\boldsymbol{x},\boldsymbol{x}^+}[f(\boldsymbol{x})^\top f(\boldsymbol{x}^+)] + \mathbb{E}_{\boldsymbol{x},\boldsymbol{x}^-}[\left((f(\boldsymbol{x})^\top f(\boldsymbol{x}^-)\right)^2], \tag{1}$$

where $f(\boldsymbol{x}) \in \mathbb{R}^p$ is the neural network representation for an input $\boldsymbol{x}$, and $\boldsymbol{x}, \boldsymbol{x}^+$ are drawn from the augmentations of the same natural data point, and $\boldsymbol{x}, \boldsymbol{x}^-$ are two augmentations generated independently either from the same data point or two different data points. The above loss function is similar to many standard contrastive loss functions (Oord et al., 2018; Sohn, 2016; Wu et al., 2018), including SimCLR (Chen et al., 2020) that we will use in our experiments. Minimizing this objective leads to representations with provable accuracy guarantees under linear probe evaluation. We use $f_{\min}$ to denote the minimizer, i.e., $f_{\min} = \arg\min_f \mathfrak{C}(f)$.

2.2. Training the Linear Head with Label Noise

After obtaining the representations of dimension p by minimizing the contrastive loss, a linear layer parameterized by $\boldsymbol{W} \in \mathbb{R}^{p\times K}$ is trained on the representations. Given a matrix $\boldsymbol{F} \in \mathbb{R}^{n\times p}$ where each row $\boldsymbol{F}_i = f_{\min}(\boldsymbol{x}_i)^\top$ is the learned representation of a data point $\boldsymbol{x}_i$, we consider the downstream task of minimizing the squared loss of the linear model on the noisy labels $\hat{\boldsymbol{Y}}$:

$$\min_{\hat{\boldsymbol{W}} \in \mathbb{R}^{p\times K}} \mathcal{L}(\hat{\boldsymbol{W}}, \boldsymbol{F}, \hat{\boldsymbol{Y}}) := \frac{1}{2}\|\hat{\boldsymbol{Y}} - \boldsymbol{F}\hat{\boldsymbol{W}}\|_F^2, \tag{2}$$

where $\hat{\boldsymbol{W}}$ is the parameters trained on the noisy labels. We apply gradient descent with a constant learning rate, η, to minimize the above loss function and denote the parameters at the t-th iteration by $\hat{\boldsymbol{W}}_t$. Formally, at iteration t we have:

$$\hat{\boldsymbol{W}}_t = \hat{\boldsymbol{W}}_{t-1} - \eta\nabla_{\hat{\boldsymbol{W}}}\mathcal{L}(\hat{\boldsymbol{W}}_{t-1}, \boldsymbol{F}, \hat{\boldsymbol{Y}}). \tag{3}$$

For simplicity, we assume $\hat{\boldsymbol{W}}_0$ is initialized as 0. While we use squared loss in our analysis, we empirically show that our results hold for other losses, such as cross-entropy.

3. Robust Learning under Extreme Noise

3.1. Contrastive learning Provably Boosts Robustness

To understand the robustness provided by contrastive learning, we build on the observation that the column space of the representation matrix spans the subspace associated with the prominent eigenvectors of the augmentation graph. Consequently, we rely on the properties of the augmentation graph to analyze the low-rank structure of the representation matrix. In particular, we utilize the following natural assumptions that formalize that: (1) the augmented data points of one sub-class are similar to each other; and (2) different from the augmented data points of other sub-classes.

Assumption 3.1 (Compact sub-class structure). For any augmented data points $\boldsymbol{x}_j$, $\boldsymbol{x}_s$ and $\boldsymbol{x}_t$ from the same sub-class, the marginal probability of $\boldsymbol{x}_s$, $\boldsymbol{x}_j$ being generated from a natural data point is close to that of $\boldsymbol{x}_t$, $\boldsymbol{x}_j$. Formally, $w_{x_s x_j}/w_{x_t x_j} \in [\frac{1}{1+\delta}, 1+\delta]$, for small $\delta \in [0, 1)$.

Assumption 3.2 (Distinguishable sub-class structure). For two pairs of augmentated examples $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $(\boldsymbol{x}_s, \boldsymbol{x}_t)$ where $\boldsymbol{x}_i, \boldsymbol{x}_j$ are from different sub-classes and $\boldsymbol{x}_s, \boldsymbol{x}_t$ are from the same sub-class, the marginal probability of $\boldsymbol{x}_i, \boldsymbol{x}_j$ being generated from a natural data is much smaller than that of $\boldsymbol{x}_s, \boldsymbol{x}_t$. Formally, $w_{x_i x_j}/w_{x_s x_t} \leq \xi$, for small $\xi \in [0,1)$.

The above assumptions result in an augmentation graph where augmented data points from different subclasses form nearly disconnected subgraphs with similar edge weights. In particular for $\xi = 0$, we get diconnected subgraph structure.

Gaussian Label Noise

We first consider the case where label noise is generated from a Gaussian distribution. This setup allows analyzing the effect of sub-class compactness δ, and distinguishability ξ on speed of overfitting, and is relevant for regression problems. Here, we assume that $\hat{\boldsymbol{Y}} = \boldsymbol{Y} + \Delta\boldsymbol{Y}$, where $\boldsymbol{Y}$ is the clean label matrix containing all the one-hot encoded labels, and $\Delta\boldsymbol{Y}$ is the label noise matrix, where each column drawn independently from $\mathcal{N}(0, \sigma^2\mathbf{I}_n/K)$.

The following theorem bounds the expected error of training the liner head on noisy labels, and explains how contrastive learning exploits the augmented sub-class structure to provably boost robustness.

Theorem 3.3. *For a dataset of size n with K classes, and $\bar{K}$ balanced compact and distinguishable sub-classes (c.f. assumptions 3.2 3.1), applying gradient descent to train a linear layer on representations obtained by minimizing contrastive loss in Eq. (1) and labels corrupted with Gaussian noise $\mathcal{N}(0, \sigma^2 \mathbf{I}_n/K)$ have the following expected error w.r.t. the ground-truth labels $\boldsymbol{Y}$:*

$$\mathbb{E}_{\Delta \boldsymbol{Y}}\left[\frac{1}{2}\|\boldsymbol{Y} - \boldsymbol{F}\hat{\boldsymbol{W}}_t\|_F^2\right] \tag{4}$$

$$\leq c_1\delta + c_2\xi + \frac{1}{2}\left(1 - \eta + \eta c_3 \xi\right)^{2t} \quad \textit{(bias)}$$

$$+ \frac{\sigma^2}{2}\left[\bar{K}\left(1 - (1-\eta)^t\right)^2 + \eta^2 c_4 (\sqrt{\delta} + \xi)^2 t^2\right], \textit{ (variance)}$$

where c_1, c_2, c_3 and c_4 are constants.

All the proofs can be found in the Appendix. We note that the above results can be easily extended to imbalanced sub-class structure.

Theorem 3.3 captures the evolution of the expected error during the training. In particular, we decompose the error into *bias* and *variance* in Eq. (4). The first part in RHS of Eq. (4) is bias and quantifies the error of the linear model trained on the clean labels of the training data. Our analysis first provides a $1 - c_3\xi$ lower-bound on the prominent singular values of the representation matrix. Then, we show that for compact and distinguishable sub-classes with small δ and ξ, the alignment of clean labels with the most prominent singular vectors corresponding to different sub-classes is lower bounded by $1 - c_1\delta - c_2\xi$. Thus, every gradient descent iteration decreases the error over all sub-classes.

The second part in RHS of Eq. (4) is the variance and captures the slowly-increasing effect of noise on the training error. As discussed, the representation matrix has one large singular vector along every sub-class, which results the first term in the variance. The second term in the variance captures the effect of contrastive learning in boosting the robustness. By bounding the smaller singular values by $\sqrt{c_4}(\sqrt{\delta} + \xi)$, we show that for a compact and distinguishable sub-class structure, with small δ and ξ, the error increases very slowly. Hence, contrastive learning substantially slows down overfitting.

Random Label Flipping

Next, we study the case where label noise $\Delta\boldsymbol{Y} = \hat{\boldsymbol{Y}} - \boldsymbol{Y}$ is generated by flipping a fraction of the clean labels at random. Formally, for a data point $\boldsymbol{x}_i$ belong to class j with $\boldsymbol{y}_i = \boldsymbol{e}_j$, if its label is flipped to class k, we have $\Delta\boldsymbol{y}_i = \boldsymbol{e}_k - \boldsymbol{e}_j$. Here, for simplicity we assume that the labels of the corrupted data points are flipped with equal probability to one of the other classes. However, our result can be trivially extended otherwise. The following theorem shows that for a dataset with compact and distinguishable sub-class structure and up to α fraction of noisy labels, the linear classifier trained on the representations obtained by contrastive learning can classify all the training data correctly. Here, we consider the case where $\xi = 0$. The general case of $\xi > 0$ requires more involved analysis which we discuss in the Appendix.

Theorem 3.4. *For a dataset with K classes and $\bar{K}$ compact and distinguishable sub-class structure (c.f. assumptions 3.2 3.1) with $\xi = 0$, let $n_{\min}, n_{\max}$ be the size of the smallest and largest sub-class, and α be the largest fraction of randomly flipped labels in any of the sub-classes. Then as long as*

$$\alpha < \frac{K-1}{K + \frac{n_{\max}}{n_{\min}} - 1} - O\left(\sqrt{\delta}\right), \tag{5}$$

applying gradient descent to train a linear classifier on representations obtained by minimizing the contrastive loss in Eq. (1), can classify all the data points correctly, i.e.,

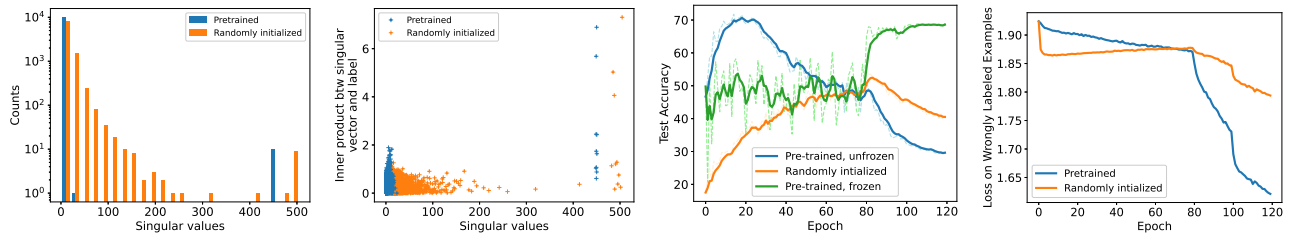
$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left[\arg\max_{j\in[K]} (\boldsymbol{F}\hat{\boldsymbol{W}})_{i,j} = \arg_{j\in[K]} (\boldsymbol{Y}_{i,j} = 1)\right] = 1.$$

Theorem 3.4 shows that contrastive learning provides robustness against a large fraction of noisy labels in the training data. The network can tolerate more noise when the sub-class structure is more compact, i.e., δ is smaller, or the sub-classes are more balanced, i.e. $n_{\max}/n_{\min} \approx 1$. In particular for $\delta = 0$, $n_{\max}/n_{\min} = 1$, we get $(K-1)/K$ noise tolerance. We note that this does not imply that a dataset with more classes necessarily has a higher tolerance. In Appendix B.1, we show that less distinguishable sub-class structure, i.e. $\xi > 0$, introduces a $O(\bar{K}^{5/2}\xi)$ perturbation to the singular values and a $O(\bar{K}^{5/2}\xi)$ rotation in the direction of singular vectors of the representation matrix. Datasets with more classes usually contains more sub-classes, which greatly reduces the noise tolerance.

In summary, contrastive learning provably boosts robustness against noisy labels by providing a representation matrix that has: (i) a prominent singular value corresponding to each subclass in the data, and significantly smaller remaining singular values; and (ii) a large alignment between the prominent singular vector and the clean labels of each subclass. The above properties allow a linear model to learn the clean labels quickly, and effectively slows down overfitting. Next, we will discuss that fine-tuning the network trained with contrastive learning on noisy labels does not overfit the noise quickly.

3.2. Fine-tuning the Network to Boost Performance

Contrastive learning slows down overfitting. In the previous section, we showed that training a linear model on representations learned by contrastive learning is provably robust. Empirically, however, we see that fine-tuning all



(a) (b) (c) (d)

Figure 1. Jacobian spectrum and dynamics of training a randomly initialized vs. fine-tuning a pre-trained ResNet32 on CIFAR-10 with 80% randomly flipped labels. (a) distribution of singular values of the Jacobian at initialization, (b) alignment of the clean label vector with the Jacobian, (c) fine-tuning a pre-trained network vs. training a randomly initialized model vs. training a linear model on representations learned by contrastive learning, and (d) loss on noisy labeled data points. While pre-training does not improve the alignment of the Jacobian and label vector, it significantly slows down overfitting by shrinking the smaller singular values of the Jacobian matrix.

layers of the deep network achieves a superior performance initially, before overfitting the noise (*c.f.* Fig. 1c and Appendix D.3).

Recall that the theoretical guarantee for linear model (theorems 3.3 and 3.4) is obtained by examining singular values and singular vectors of $\boldsymbol{F}$. Here, we use a similar idea to understand benefits of contrastive learning on robustness when all the layers are trained. Intuitively, during the early stage of training, it is natural to assume that the gradient does not considerably change, and therefore the model is nearly linear. Here, the initial Jacobian matrix plays the same role as the representation matrix, $\boldsymbol{F}$, to the linear model.

We examine the spectrum of the Jacobian of a network pre-trained with contrastive learning and compare it to that of a randomly initialized network. Fig. 1a, 1b compares the Jacobian spectrum and alignment of clean label vector with Jacobian of ResNet32 on a random sample of 1000 data points from CIFAR10. Interestingly, Fig. 1b shows that while pre-training does not considerably improve (in Appendix C we show a slight improvement) the alignment between singular vectors of the Jacobian and the clean label vector, it greatly shrinks the smaller singular values of the Jacobian, as is illustrated by Fig. 1a. As a result, it takes substantially longer for the pre-trained network to overfit the noisy labels. As Fig. 1d shows, while a randomly initialized network experience a sharp drop in loss of noisy labeled data points during the first few epochs of training, it takes much longer for a pre-trained network to overfit the noise. In particular, Fig. 1c demonstrates the superior performance of a pre-trained network over that of a randomly initialized and a linear model trained on representations learned by contrastive learning, before it overfits the noise.

Contrastive learning boosts robust methods. As discussed, pre-training the network with contrastive learning effectively shrinks the smaller singular values of the Jacobian and slows down overfitting the noisy labels. Our experiments (*c.f.* Table 1 and Appendix D) show that the initial level of robustness provided by contrastive learning can be leveraged by existing robust training methods to achieve a superior performance under extreme noise levels. In particular, we consider three methods that prevent overfitting through regularization, namely, Mixup (Zhang et al., 2017) and ELR (Liu et al., 2020), or identifying clean examples, namely, CRUST (Mirzasoleiman et al., 2020).

Table 1. Test accuracy on mini WebVision. The best test accuracy is marked in bold. SimCLR (CL) pre-training leads to average improvements of 4.11% and 3.20% for mini Webvision and ImageNet respectively. E2E refers to end to end fine-tuning the pre-trained network.

Method	WebVision		ImageNet	
	Top-1	Top-5	Top-1	Top-5
Crust	72.40	89.56	67.36	87.84
Mixup	71.38	87.36	68.34	88.44
ELR	76.26	91.26	68.71	87.84
CL + E2E	71.84	88.84	68.48	89.32
CL + Mixup	76.34	90.52	**72.25**	**89.72**
CL + ELR	**79.52**	**93.80**	71.20	90.80

4. Conclusion

We showed that representations learned by contrastive learning provably boosts robustness against noisy labels. In particular, contrastive learning provides a representation matrix that has: (i) a significant gap between largest and smallest singular values; (ii) a large alignment between the prominent singular vectors and the clean labels. The above properties allow a linear layer trained on the embeddings to learn the clean labels quickly with relatively slow overfitting. Then we extend this idea to explain the observation that training end to end also has a good performance at the early stage of learning. Finally, we demonstrated that the initial robustness provided by contrastive learning enables state-of-the-art robust methods to achieve a superior performance under extreme noise levels. Our results confirm benefits of contrastive learning for robust training.

References

Chen, P., Liao, B. B., Chen, G., and Zhang, S. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pp. 1062–1070, 2019.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Eckart, C. and Young, G. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Ghosh, A. and Lan, A. Contrastive learning improves model robustness under label noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2703–2708, 2021.

Han, B., Yao, Q., Yu, X., Niu, G., Xu, M., Hu, W., Tsang, I., and Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pp. 8527–8537, 2018.

HaoChen, J. Z., Wei, C., Gaidon, A., and Ma, T. Provable guarantees for self-supervised deep learning with spectral contrastive loss. *arXiv preprint arXiv:2106.04156*, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. Using self-supervised learning can improve model robustness and uncertainty. *arXiv preprint arXiv:1906.12340*, 2019.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*, pp. 2309–2318, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Li, W., Wang, L., Li, W., Agustsson, E., and Van Gool, L. Webvision database: Visual learning and understanding from web data. *arXiv preprint arXiv:1708.02862*, 2017.

Liu, S., Niles-Weed, J., Razavian, N., and Fernandez-Granda, C. Early-learning regularization prevents memorization of noisy labels. *arXiv preprint arXiv:2007.00151*, 2020.

Ma, X., Wang, Y., Houle, M. E., Zhou, S., Erfani, S., Xia, S., Wijewickrema, S., and Bailey, J. Dimensionality-driven learning with noisy labels. In *International Conference on Machine Learning*, pp. 3355–3364, 2018.

Malach, E. and Shalev-Shwartz, S. Decoupling" when to update" from" how to update". In *Advances in Neural Information Processing Systems*, pp. 960–970, 2017.

Minc, H. On the maximal eigenvector of a positive matrix. *SIAM Journal on Numerical Analysis*, 7(3):424–427, 1970.

Mirzasoleiman, B., Cao, K., and Leskovec, J. Coresets for robust training of deep neural networks against noisy labels. *Advances in Neural Information Processing Systems*, 33, 2020.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. Automatic differentiation in pytorch. 2017.

Patrini, G., Rozza, A., Krishna Menon, A., Nock, R., and 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*, pp. 1944–1952, 2017.

SimCLR. https://github.com/spijkervet/simclr.

Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in neural information processing systems*, pp. 1857–1865, 2016.

Stewart, G. W. Matrix perturbation theory. 1990.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. A. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

Wedin, P.-Å. Perturbation bounds in connection with singular value decomposition. *BIT Numerical Mathematics*, 12(1):99–111, 1972.

Wu, Z., Xiong, Y., Yu, S. X., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., and Sugiyama, M. Are anchor points really indispensable in label-noise learning? In *Advances in Neural Information Processing Systems*, pp. 6838–6849, 2019.

Xu, Y., Cao, P., Kong, Y., and Wang, Y. L_dmi: A novel information-theoretic loss function for training deep nets robust to label noise. In *Advances in Neural Information Processing Systems*, pp. 6225–6236, 2019.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A. M., and Litany, O. Contrast to divide: Self-supervised pre-training for learning with noisy labels. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1657–1667, 2022.

A. Analysis for Disconnected Subclasses

In this section we consider the case where $\xi = 0$ in assumption 3.2, which implies that the probability of two augmentation data from different subclasses being generated from the same random natural datum is exactly zero. And in section B we extend the results to any $\xi \in [0, 1)$ via eigenvalue and eigenvector perturbation. We use $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\|_F$ to denote the 1-norm, operator norm and Frobenius norm, respectively.

A.1. Spectral Decomposition of Augmentation Graph

An important technical idea we use to formalize the representations obtained by contrastive learning is augmentation graph (HaoChen et al., 2021), which is an undirected graph with all augmentation data $\{x_1, x_2, \ldots, x_n\}$ as its vertices and $w_{x_i x_j}$ as the weight for edge (x_i, x_j). Let A denote the adjacency matrix of the augmentation graph, that is, each entry $a_{ij} = w_{x_i x_j}$. And the normalized adjacency matrix $\overline{A}$ is defined as

$$\overline{A} := D^{-1/2} A D^{-1/2},$$

where $D = \text{diag}(w_{x_1}, w_{x_2}, \ldots, w_{x_n})$ with $w_{x_i} = \sum_{j=1}^{n} w_{x_i x_j}$. For simplicity we index the augmentation data in the following way: the first n_1 data are from subclass 1, the next n_2 data are from subclass 2, ..., the last $n_{\bar{K}}$ data are from subclass $\bar{K}$. Lemma A.1 states an important property of $\overline{A}$.

Lemma A.1. *Assumption 3.2 with $\xi = 0$ implies that $\overline{A}$ is a block diagonal matrix diag$(\overline{A}_1, \overline{A}_2, \ldots, \overline{A}_{\bar{K}})$ where $\overline{A}_{\bar{k}} \in \mathbb{R}^{n_{\bar{k}} \times n_{\bar{k}}}$. Denote the entries of $\overline{A}_{\bar{k}}$ by $\bar{a}_{k,i,j}$. Combining that and 3.1 yields: for each block $\overline{A}_{\bar{k}}$, the ratio between two entries in the same column is bounded as follows*

$$\forall \bar{k} \in [\bar{K}], \max_{j,s,t} \frac{\bar{a}_{\bar{k},s,j}}{\bar{a}_{\bar{k},t,j}} \leq 1 + \delta',$$

where $\delta' = (1 + \delta)^{3/2} - 1 = O(\delta)$.

Let $f_{\min} = \arg\min_f \mathfrak{C}(f)$ and $\boldsymbol{F}_{\min} = [f_{\min}(x_1)\ f_{\min}(x_2) \ldots f_{\min}(x_n)]^\top$, according the theorem in (HaoChen et al., 2021), $\boldsymbol{F}_{\min}$ is also the minimizer of the following matrix factorization problem

$$\min_{\boldsymbol{F} \in \mathbb{R}^{n \times p}} \|\overline{A} - \boldsymbol{F}\boldsymbol{F}^\top\|_F^2, \tag{6}$$

and therefore can be further decomposed as

$$\boldsymbol{F}_{\min} = F^* \Sigma R, \tag{7}$$

by Eckart–Young–Mirsky theorem (Eckart & Young, 1936), where $F^* \in \mathbb{R}^{n \times p} = [v_1, v_2, \ldots, v_p] \in \mathbb{R}^{n \times p}$, $\Sigma = \text{diag}(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_p})$, $R \in \mathbb{R}^{p \times p}$ is some orthogonal matrix, $\lambda_1, \lambda_2, \ldots, \lambda_p$ are the p largest eigenvalues of $\overline{A}$ and $v_1, v_2, \ldots, v_p$ are the corresponding unit-norm eigenvectors. Our following proofs are all based on this decomposition. To avoid cluttered notation we use $\boldsymbol{F}$ instead of $\boldsymbol{F}_{\min}$ to denote the minimizer.

A.2. Error under Gaussian Noise when $\xi = 0$

By the rule of gradient descent 3 and the decomposition of $\boldsymbol{F}$ 7, the output of the linear model at the t-th iteration on the training data can be written as

$$\boldsymbol{F}\hat{\boldsymbol{W}}_t = F^* B_t {F^*}^\top (\boldsymbol{Y} + \Delta\boldsymbol{Y}), \tag{8}$$

where $B_t = \text{diag}(\beta_1, \beta_2, \ldots, \beta_p)$ with $\beta_i = 1 - (1 - \eta\lambda_i)^t$.

First we rewrite the error in terms of the eigenvalues and eigenvectors of $\overline{A}$

$$\begin{aligned}\mathbb{E}_{\Delta \boldsymbol{Y}}\mathcal{L}(\hat{\boldsymbol{W}}_t, \boldsymbol{F}, \boldsymbol{Y}) =& \mathbb{E}_{\Delta \boldsymbol{Y}}\left[\frac{1}{2}\|\boldsymbol{Y} - F^* B_t F^{*\top}(\boldsymbol{Y} + \Delta \boldsymbol{Y})\|_F^2\right] \\ =& \frac{1}{2}\|\boldsymbol{Y} - F^* B_t F^{*\top}\boldsymbol{Y}\|_F^2 + \mathbb{E}_{\Delta \boldsymbol{Y}}\left[\|F^* B_t F^{*\top}\Delta \boldsymbol{Y}\|_F^2\right] \\ =& \frac{1}{2}\left(\|\boldsymbol{Y}\|_F^2 + \left\langle \boldsymbol{Y}, F^* B_t^2 F^{*\top}\boldsymbol{Y}\right\rangle_F - 2\left\langle \boldsymbol{Y}, F^* B_t F^{*\top}\boldsymbol{Y}\right\rangle_F\right) + \frac{\sigma^2}{2}\|F^* B_t F^{*\top}\|_F^2 \\ =& \underbrace{\frac{n}{2} + \frac{1}{2}\sum_{i=1}^{p}(\beta_i^2 - 2\beta_i)\sum_{j=1}^{K}(v_i^T y_j)^2}_{\text{bias}} + \underbrace{\frac{\sigma^2}{2}\sum_{i=1}^{p}\beta_i^2}_{\text{variance}},\end{aligned} \tag{9}$$

where $y_j \in \mathbb{R}^n$ is the j-th *column* in $\boldsymbol{Y}$.

For each block $\overline{A}_{\bar{k}}$, let $\lambda_{\bar{k},1}, \lambda_{\bar{k},2}, \ldots, \lambda_{\bar{k},n_{\bar{k}}}$ denote the eigenvalues (in descending order) and $v_{\bar{k},1}, v_{\bar{k},2}, \ldots, v_{\bar{k},n_{\bar{k}}}$ denote the corresponding eigenvectors. Then the eigenvalues for $\overline{A}$ are the list of the eigenvalues of all blocks. And the corresponding eigenvectors are the block vectors $(\vec{0}_1, \vec{0}_2, \ldots, \vec{0}_{\bar{k}-1}, v_{\bar{k},i}, \vec{0}_{\bar{k}+1}, \ldots, \vec{0}_{\bar{K}}) := \dot{v}_{\bar{k},i}$ with each $\vec{0}_j$ being a vector consists of n_j zeros. Since $\overline{A}$ is a normalized adjacency matrix, each block $\overline{A}_{\bar{k}}$ is also normalized. Then the largest eigenvalue for each block equals to 1, i.e., $\lambda_{\bar{k},1} = 1$. As long as $p \geq \bar{K}$, all $\lambda_{\bar{k},1}$ and $v_{\bar{k},1}$ are selected to compose $\boldsymbol{F}$. Let $p_{\bar{k}} \geq 1$ be the number of eigenvalues/eigenvectors selected from each block $\overline{A}_{\bar{k}}$. Note that $\beta_i^2 - 2\beta_i \leq 0$, since $\beta_i \in [0, 1]$, we get the following bound for the second term of the bias in equation 9

$$\begin{aligned}\sum_{i=1}^{p}(\beta_i^2 - 2\beta_i)\sum_{j=1}^{K}(v_i^\top y_j)^2 \leq& \sum_{\bar{k}=1}^{\bar{K}}(\beta_{\bar{k},1}^2 - 2\beta_{\bar{k},1})\sum_{j=1}^{K}(\dot{v}_{\bar{k},1}^\top y_j)^2 \\ \leq& -\sum_{\bar{k}=1}^{\bar{K}}(1 - (1-\eta)^{2t})\sum_{j=1}^{K}(\dot{v}_{\bar{k},1}^\top y_j)^2.\end{aligned} \tag{10}$$

By Perron-Frobenious theorem all elements in $v_{\bar{k},1}$ are positive. With the observation that $\dot{v}_{\bar{k},1}^T y_j = \|v_{\bar{k},1}\|_1$ only when $\bar{k}$ is a subclass of class j and otherwise 0, equation 10 can be rewritten as

$$\sum_{i=1}^{p}(\beta_i^2 - 2\beta_i)\sum_{j=1}^{K}(v_i^\top y_j)^2 \leq -(1 - (1-\eta)^{2t})\sum_{\bar{k}=1}^{\bar{K}}\|v_{\bar{k},1}\|_1^2. \tag{11}$$

Then we introduce the following lemma

Lemma A.2. *By assumption 3.1, the* 1*-norm of* $v_{\bar{k},1}$ *has the following lower bound.*

$$\|v_{\bar{k},1}\|_1^2 \geq \frac{n_{\bar{k}}^2}{1 + (n_{\bar{k}} - 1)(1 + \delta')^2}.$$

Proof. Write $v_{\bar{k},1}$ as $[e_1, e_2, \ldots, e_{n_{\bar{k}}}]^T$, the elements of which are all positive since $\overline{A}_{\bar{k}}$ is a positive matrix, by Perron-Frobenius theorem. Then the quotient of two elements in $v_{\bar{k},1}$ can be bounded in terms of the entries of $\overline{A}_{\bar{k}}$ (Minc, 1970) and therefore $1 + \delta'$ by lemma A.1

$$\max_{i,j}\frac{e_i}{e_j} \leq \max_{j,s,t}\frac{\bar{a}_{\bar{k},s,j}}{\bar{a}_{\bar{k},t,j}} \leq 1 + \delta'.$$

Then, let $e_{min} = \min_i e_i$, we have

$$1 = \|v_{\bar{k},1}\|_2^2 = \sum_{i=1}^{n_{\bar{k}}} e_i^2 \leq e_{min}^2 + (n_{\bar{k}} - 1)(1 + \delta')^2 e_{min}^2,$$

and

$$\|v_{\bar{k},1}\|_1^2 = (\sum_{i=1}^{n_{\bar{k}}} e_i)^2 \geq n_{\bar{k}}^2 e_{min}^2.$$

Combining the preceding two equations yields

$$\|v_{\bar{k},1}\|_1^2 \geq \frac{n_{\bar{k}}^2}{1+(n_{\bar{k}}-1)(1+\delta')^2}.$$

□

Now we are ready to bound the bias term by combining equation 11 and lemma A.2.

$$\begin{aligned}
2 \times \text{bias} \leq & n - (1-(1-\eta)^{2t}) \sum_{\bar{k}=1}^{\bar{K}} \frac{n_{\bar{k}}^2}{1+(n_{\bar{k}}-1)(1+\delta')^2} \\
= & \sum_{\bar{k}=1}^{\bar{K}} n_k \left(1 - \frac{n_k}{1+(n_k-1)(1+\delta')^2}\right) + (1-\eta)^{2t} \sum_{\bar{k}=1}^{\bar{K}} \frac{n_{\bar{k}}^2}{1+(n_{\bar{k}}-1)(1+\delta')^2} \\
\leq & \sum_{\bar{k}=1}^{\bar{K}} \frac{n_{\bar{k}}}{\frac{n_{\bar{k}}}{(\delta'^2+2\delta')(n_{\bar{k}}-1)}+1} + (1-\eta)^{2t} n \\
= & 2(n-\bar{K})O(\delta) + (1-\eta)^{2t} n
\end{aligned}$$

Next, we introduce lemma A.3

Lemma A.3. *The sum of squared eigenvalues for each block $\bar{k}$ can be bounded.*

$$\sum_{i=1}^{n_{\bar{k}}} \lambda_{\bar{k},i}^2 \leq \frac{(1+(n_{\bar{k}}-1)(1+\delta')^2)(1+\delta')^2}{n_{\bar{k}}}.$$

Proof. First we have

$$\sum_{i=1}^{n_{\bar{k}}} \lambda_{\bar{k},i}^2 = \|\overline{A}_{\bar{k}}\|_F^2 = \sum_{j=1}^{n_{\bar{k}}} \|c_{\bar{k},j}\|_2^2, \tag{12}$$

where $c_{\bar{k},j} = [\bar{a}_{\bar{k},1,j}, \bar{a}_{\bar{k},2,j}, \ldots, \bar{a}_{\bar{k},n_{\bar{k}},j}]^T$ denotes the j-th column in $\overline{A}_{\bar{k}}$. Analogous to lemma A.2, here we can bound $\|c_{\bar{k},j}\|_2^2$ in terms of $\|c_{\bar{k},j}\|_1^2$ by lemma A.1

$$\|c_{\bar{k},j}\|_2^2 \leq \frac{(1+(n_{\bar{k}}-1)(1+\delta')^2)\|c_{\bar{k},j}\|_1^2}{n_k^2}. \tag{13}$$

We also have

$$\|c_{\bar{k},j}\|_1^2 \leq \max_j \|c_{\bar{k},j}\|_1^2 \leq (1+\delta')^2 \min_j \|c_{\bar{k},j}\|_1^2 \leq (1+\delta')^2 \lambda_{\bar{k},1}^2 = (1+\delta')^2, \tag{14}$$

where the second inequality holds because of assumption 3.1 and the third inequality holds because of Perron-Frobenius theorem. Combining equations 12, 13 and 14 we get lemma A.3 □

Lemma A.4. *Besiedes the $\bar{K}$ largest eigenvalues that are all equal to 1, each of the remaining eigenvalues is upper bounded by*

$$\lambda_{\bar{k},i} \leq \sqrt{\frac{(1+(n_{\bar{k}}-1)(1+\delta')^2)(1+\delta')^2}{n_{\bar{k}}} - 1} = O(\sqrt{\delta}), \quad \forall i = 2,3,\ldots,n_{\bar{k}}, \quad \forall \bar{k} \in [\bar{K}].$$

Proof. By lemma A.3 we know that

$$\begin{aligned}\sum_{i=2}^{p_{\bar{k}}} \lambda_{\bar{k},i}^2 \leq & \sum_{i=2}^{n_{\bar{k}}} \lambda_{\bar{k},i}^2 \\ = & \sum_{i=1}^{n_{\bar{k}}} \lambda_{\bar{k},i}^2 - \lambda_{\bar{k},1}^2 \\ \leq & \frac{(1+(n_{\bar{k}}-1)(1+\delta')^2)(1+\delta')^2}{n_{\bar{k}}} - 1\end{aligned}$$

Then

$$\begin{aligned}\lambda_{\bar{k},i} \leq & \sqrt{\sum_{i=2}^{n_{\bar{k}}} \lambda_{\bar{k},i}^2} \\ \leq & \sqrt{\frac{(1+(n_{\bar{k}}-1)(1+\delta')^2)(1+\delta')^2}{n_{\bar{k}}} - 1}\end{aligned}$$

□

Now we are ready to bound the variance term in 9.

$$\begin{aligned}\sum_{i=1}^{p} \beta_i^2 = & \sum_{\bar{k}=1}^{\bar{K}} \sum_{i=1}^{p_{\bar{k}}} \beta_{\bar{k},i}^2 \\ = & \sum_{\bar{k}=1}^{\bar{K}} \beta_{\bar{k},1}^2 + \sum_{\bar{k}=1}^{\bar{K}} \sum_{i=2}^{p_{\bar{k}}} \beta_{\bar{k},i}^2 \\ = & \bar{K}(1-(1-\eta)^t)^2 + \sum_{\bar{k}=1}^{\bar{K}} \sum_{i=2}^{p_{\bar{k}}} \beta_{\bar{k},i}^2 \\ \leq & \bar{K}(1-(1-\eta)^t)^2 + \sum_{\bar{k}=1}^{\bar{K}} (p_{\bar{k}}-1)\beta_{\bar{k},2}^2. \end{aligned} \tag{15}$$

Recalling that $\beta_{\bar{k},2} = 1-(1-\eta\lambda_{\bar{k},2})^t$ and plugging lemma A.4 into equation 15 yield

$$\sum_{i=1}^{p} \beta_i^2 \leq \bar{K}(1-(1-\eta)^t)^2 + (p-\bar{K})\left(1-(1-\eta O(\sqrt{\delta}))^t\right)^2.$$

Plugging the preceding into the variance term yields

$$\text{variance} \leq \frac{\sigma^2}{2}\bar{K}\left(1-(1-\eta)^t\right)^2 + \frac{\sigma^2}{2}(p-\bar{K})\left(1-\left(1-\eta O(\sqrt{\delta})\right)^t\right)^2.$$

A.3. Accuracy under Label Flipping (Proof for Theorem 3.4)

Define $\pi_{\bar{k}}(\delta')$ and $\rho_{\bar{k}}(\delta')$ as

$$\begin{cases}\pi_{\bar{k}}(\delta') := \frac{n_{\bar{k}}}{1+(n_{\bar{k}}-1)(1+\delta')^2}, \\ \rho_{\bar{k}}(\delta') := (1+\delta')^2/\pi_{\bar{k}}(\delta') - 1\end{cases}$$

We study the accuracy by looking at the entries of the output $\boldsymbol{F}\hat{\boldsymbol{W}}_t$.

$$
\begin{aligned}
\boldsymbol{F}\hat{\boldsymbol{W}}_t &= F^* B_t F^{*\top}(\boldsymbol{Y} + \Delta\boldsymbol{Y}) \\
&= \left[F^* B_t F^{*\top}(y_1 + \Delta y_1),\ F^* B_t F^{*\top}(y_2 + \Delta y_2),\ \ldots,\ F^* B_t F^{*\top}(y_K + \Delta y_K)\right] \\
&= \left[\sum_{i=1}^{p} \beta_i v_i v_i^\top (y_1 + \Delta y_1),\ \sum_{i=1}^{p} \beta_i v_i v_i^\top (y_2 + \Delta y_2),\ \ldots,\ \sum_{i=1}^{p} \beta_i v_i v_i^\top (y_K + \Delta y_K)\right] \\
&:= [z_1, z_2, \ldots, z_K]
\end{aligned}
$$

For simplicity we introduce the notations C_k and $S_{\bar{k}}$, which are the sets of indices of data from class k and subclass $\bar{k}$, respectively

$$
\begin{aligned}
C_k &:= \{i : \text{datum } x_i \text{ belongs to class } k\} \\
S_{\bar{k}} &:= \{i : \text{datum } x_i \text{ belongs to subclass } \bar{k}\} = \{i : \sum_{j=1}^{\bar{k}-1} n_j < i \leq \sum_{j=1}^{\bar{k}} n_j\}.
\end{aligned}
$$

Let $\mu^{(j)}$ denote the j-th element of vector μ. Then $z_k^{(j)} = \sum_{i=1}^{p} \beta_i v_i^{(j)} v_i^\top (y_k + \Delta y_k)$. Let $\bar{k}_j$ denote the subclass that x_j belongs to, i.e., $j \in S_{\bar{k}}$ and define $e_{\min,s} := \min_j v_{\bar{k}_j,1}^{(s)}$ and $e_{\max,s} := \max_s v_{\bar{k}_j,1}^{(s)}$. We have the following two lemmas.

Lemma A.5. $z_k^{(j)}$ *can be bounded*

$$
\begin{cases}
z_k^{(j)} \geq & (1-(1-\eta)^t)\, e_{\min,j} \dot{v}_{\bar{k}_j,1}^{(j)} n_{\min}(1-\alpha) - \sqrt{n_{\max}}(p - \bar{K} + 1)\left(1 - (1 - \eta\sqrt{\rho_{\bar{k}_{\max}}(\delta')})^t\right), \quad j \in C_k \\
z_k^{(j)} \leq & (1-(1-\eta)^t)\, e_{\max,j} \dot{v}_{\bar{k}_j,1}^{(j)} n_{\max} \frac{\alpha}{K-1} + \sqrt{\frac{n_{\max}}{K-1}}(p - \bar{K} + 1)\left(1 - (1 - \eta\sqrt{\rho_{\bar{k}_{\max}}(\delta')})^t\right), \quad j \notin C_k.
\end{cases}
$$

Proof. Recalling that one property of the block vector $\dot{v}_{\bar{k},i}$ is that $\dot{v}_{\bar{k},i}^{(j)} = 0$ when $j \notin S_{\bar{k}}$, we have

$$
\begin{aligned}
z_k^{(j)} &= \sum_{i=1}^{p} \beta_i v_i^{(j)} v_i^\top (y_k + \Delta y_k) \\
&= \sum_{\bar{k}=1}^{\bar{K}} \sum_{i=1}^{p_{\bar{k}}} \beta_{\bar{k},i} \dot{v}_{\bar{k},i}^{(j)} \dot{v}_{\bar{k},i}^\top (y_k + \Delta y_k) \\
&= \sum_{i}^{p_{\bar{k}_j}} \beta_{\bar{k}_j,i} \dot{v}_{\bar{k}_j,i}^{(j)} \dot{v}_{\bar{k}_j,i}^\top (y_k + \Delta y_k) \\
&= \beta_{\bar{k}_j,1} \dot{v}_{\bar{k}_j,1}^{(j)} \dot{v}_{\bar{k}_j,1}^\top (y_k + \Delta y_k) + \sum_{i=2}^{p_{\bar{k}_j}} \beta_{\bar{k}_j,i} \dot{v}_{\bar{k}_j,i}^{(j)} \dot{v}_{\bar{k}_j,i}^\top (y_k + \Delta y_k) \\
&= \left(1 - (1-\eta)^t\right) \dot{v}_{\bar{k}_j,1}^{(j)} \dot{v}_{\bar{k}_j,1}^\top (y_k + \Delta y_k) + \sum_{i=2}^{p_{\bar{k}_j}} \beta_{\bar{k}_j,i} \dot{v}_{\bar{k}_j,i}^{(j)} \dot{v}_{\bar{k}_j,i}^\top (y_k + \Delta y_k)
\end{aligned} \tag{16}
$$

If $j \in C_k$, there are at least $n_{\min}(1-\alpha)$ elements being 1 with others being 0 in $y_k + \Delta y_k$; if $j \notin C_k$, there are at most $n_{\max}\frac{\alpha}{K-1}$ elements being 1 with others being 0 in $y_k + \Delta y_k$.Then the inner product in the first term in equation 16 can be bounded by

$$
\dot{v}_{\bar{k}_j,1}^\top (y_k + \Delta y_k) = \sum_{s=1}^{n_{\bar{k}_j}} \dot{v}_{\bar{k}_j,1}^{(s)} (y_k + \Delta y_k)^{(s)} \begin{cases} \geq e_{\min,j} n_{\min}(1-\alpha), & j \in C_k \\ \leq e_{\max,j} \dot{v}_{\bar{k}_j,1}^{(j)} n_{\max} \frac{1-\alpha}{K-1}, & j \notin C_k \end{cases}
$$

For the inner product in the second term in equation 16, if $j \in C_k$, $\dot{v}_{\bar{k}_j,i}^\top (y_k + \Delta y_k)$ is the sum of at most $n_{\max}$ elements in $\dot{v}_{\bar{k}_j,i}$, since $\|\dot{v}_{\bar{k}_j,i}\|_2^2 = 1$, the sum is bounded by $[-\sqrt{n_{\max}}, \sqrt{n_{\max}}]$; similarly we get the bound

$[-\sqrt{n_{\max}/(K-1)}, \sqrt{n_{\max}/(K-1)}]$ for $j \notin C_k$. Also we know that $\|\dot{v}^{(j)}_{\bar{k}_j,i}\| \leq 1$. Then it remains to bound $\sum_{i=2}^{p_{\bar{k}_j}} \beta_{\bar{k}_j,i}$. By lemma A.4, we have

$$\begin{aligned}\sum_{i=2}^{p_{\bar{k}_j}} \beta_{\bar{k}_j,i} \leq & (p_{\bar{k}_j} - 1)\beta_{\bar{k}_j,2} \\ \leq & (p - \bar{K} + 1)\beta_{\bar{k}_j,2} \\ \leq & (p - \bar{K} + 1)\beta_{\bar{k}_j,2} \\ \leq & (p - \bar{K} + 1)\left(1 - (1 - \eta\sqrt{\rho_{\bar{k}_{\max}}(\delta')})^t\right)\end{aligned}$$

□

Lemma A.6. $e_{\min,j}$, $e_{\max,j}$ *and* $e_{max,j}/e_{\min,j}$ *can be bounded.*

$$\begin{aligned}e_{\min,j} &\geq \sqrt{\frac{1}{1 + (n_{\max} - 1)(1+\delta')^2}} \\ e_{\max,j} &\leq \sqrt{\frac{1}{1 + (n_{\min} - 1)/(1+\delta')^2}} \\ \frac{e_{\max,j}}{e_{\min,j}} &\leq (1+\delta')\sqrt{\frac{(1 + (1+\delta')^2(n_{\max} - 1))}{n_{\max} - 1 + (1+\delta')^2}} := \Gamma(\delta')\end{aligned}$$

The proof is analogous to that for lemma A.2.

Let $z_k^{(j:j\in C_k)} > z_k^{(i:i\notin C_k)}$, then by lemma A.5 we get

$$\alpha < \frac{(1 - \frac{1}{e_{\min,j}\dot{v}^{(j)}_{\bar{k}_j,1} n_{\min}}(1 + \frac{1}{\sqrt{K}})\Psi(\delta'))(K-1)}{K - 1 + \frac{n_{\max}e_{\max,j}}{n_{\min}e_{\min,j}}}, \tag{17}$$

where

$$\begin{aligned}\Psi(\delta') =& \frac{\sqrt{n_{\max}}(p - \bar{K} + 1)\left(1 - (1 - \eta\sqrt{\rho_{\bar{k}_{\max}}(\delta')})^t\right)}{1 - (1-\eta)^t} \\ =& O(\frac{1 - (1 - \eta\sqrt{\rho_{\bar{k}_{\max}}(\delta')})^t}{1 - (1-\eta)^t}).\end{aligned}$$

Plugging lemma A.6 into equation 17 with some algebraic manipulation yields

$$\alpha < \frac{K-1}{K + \frac{n_{\max}}{n_{\min}} - 1} - O(\frac{1 - (1 - \eta\sqrt{\rho_{\bar{k}_{\max}}(\delta')})^t}{1 - (1-\eta)^t}).$$

An interesting observation is that $\frac{1-(1-\eta\sqrt{\rho_{\bar{k}_{\max}}(\delta')})^t}{1-(1-\eta)^t}$ monotonically increases with t since we assume δ' is very small, meaning that for there to exist a t such that the preceding inequality holds, it must hold when $t = 1$. Then substituting $t = 1$ into the right-hand-side yields

$$\begin{aligned}\alpha <& \frac{K-1}{K + \frac{n_{\max}}{n_{\min}} - 1} - O(\sqrt{\rho_{\bar{k}_{\max}}(\delta')}) \\ =& \frac{K-1}{K + \frac{n_{\max}}{n_{\min}} - 1} - O(\sqrt{\delta}).\end{aligned}$$

It is worth mentioning that we didn't get this requirement on noise level by specifying $t = 1$. Instead we want to find a sufficient condition for there to exist an epoch t s.t. the model $\hat{\boldsymbol{W}}_t$ predicts the ground-truth labels for all training examples. And the last step in the proof shows that this sufficient condition holds if and only if it holds for $t = 1$.

B. Considering Off-Diagonal Entries in the Adjacency Matrix (Connected Subclasses)

For here on we assume $n_{\bar{k}} = \frac{n}{\bar{K}}, \forall \bar{k} \in \bar{K}$ for simplicity, despite that our results can easily extend to unbalanced dataset.

Lemma B.1. *Under assumption 3.2, the off-diagonal entries in A is no longer zero. Let $\widetilde{A}$ denote the new normalized matrix, which also contains non-zero off-diagonal entries. With a bit abuse of notation, in the following we use $\overline{A}$ to denote the matrix obtained by normalizing A with off-diagonal elements ignored. Then all the properties of eigenvectors and eigenvalues of $\overline{A}$ stated before including those lemma A.1, A.2, A.3 still hold. And $\widetilde{A}$ can be written as a perturbation of $\overline{A}$*

$$\widetilde{A} = \overline{A} + E,$$

with $\|E\|_F = O\left(\bar{K}^{5/2}\xi\right)$.

Proof. Let $\mathring{A}$ be a matrix in the same shape of $\overline{A}$ containing all elements of $\widetilde{A}$ in the diagonal blocks. Let H be a matrix that keep the remaining off-diagonal elements. Therefore $\widetilde{A} = \mathring{A} + H$, which can be rewritten as

$$\widetilde{A} = \overline{A} + \mathring{A} - \overline{A} + H \tag{18}$$

For all off-diagonal elements $h_{i,j}$ in H

$$h_{i,j} \leq \frac{\bar{K}\xi}{n}.$$

Since there are $n^2 - \sum_{\bar{k}=1}^{\bar{K}} n_{\bar{k}}^2$ entries outside of the blocks, the norm of H can be bounded by

$$\|H\|_F \leq \sqrt{1 - \frac{1}{\bar{K}}}\bar{K}\xi \tag{19}$$

Each element in the diagonal blocks of $\overline{A} - \mathring{A}$ is non-negative. Also, supposing $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are from subclass $\bar{k}$, we have

$$\begin{aligned}
(\overline{A} - \mathring{A})_{i,j} =& \frac{w_{x_i x_j}}{\sqrt{\sum_{s:\boldsymbol{x}_s \in C_{\bar{k}}} w_{x_i x_s}}\sqrt{\sum_{t:\boldsymbol{x}_t \in C_{\bar{k}}} w_{x_t x_j}}} - \frac{w_{x_i x_j}}{\sqrt{\sum_{s=1}^{n} w_{x_i x_s}}\sqrt{\sum_{t=1}^{n} w_{x_t x_j}}} \\
\leq& \frac{n(\bar{K}-1)\xi}{\frac{n}{\bar{K}(1+\delta)}(\frac{n}{\bar{K}(1+\delta)} + n(\bar{K}-1)\xi)} \\
=& O\left(\frac{\bar{K}^3(1+\delta)^2\xi}{n}\right),
\end{aligned}$$

by which the norm of $\overline{A} - \mathring{A}$ is bounded

$$\|\overline{A} - \mathring{A}\|_F = O\left(\bar{K}^{5/2}(1+\delta)^2\xi\right) = O\left(\bar{K}^{5/2}\xi\right). \tag{20}$$

Combining equations 18, 19 and 20 completes the proof. □

$\overline{A}$ has the following eigendecomposition

$$\overline{A} = \begin{bmatrix} V_I V_N \end{bmatrix} \begin{bmatrix} \Sigma_I & 0 \\ 0 & \Sigma_N \end{bmatrix} \begin{bmatrix} V_I^\top \\ V_N^\top \end{bmatrix},$$

where Σ_I collects the $\bar{K}$ largest eigenvalues $\lambda_1, \ldots, \lambda_{\bar{K}}$ on the diagonal and Σ_N collects the remaining $\lambda_{\bar{K}+1}, \ldots, \lambda_{\bar{n}}$. V_I and V_N collects the corresponding $\bar{K}$ and $n - \bar{K}$ eigenvectors, respectively. Let $\widetilde{A}$ has analogous decomposition

$$\widetilde{A} = \begin{bmatrix} \widetilde{V}_I \widetilde{V}_N \end{bmatrix} \begin{bmatrix} \widetilde{\Sigma}_I & 0 \\ 0 & \widetilde{\Sigma}_N \end{bmatrix} \begin{bmatrix} \widetilde{V}_I^\top \\ \widetilde{V}_N^\top \end{bmatrix},$$

with eigenvalues $\widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_n$ and eigenvectors $\widetilde{v}_1, \ldots, \widetilde{v}_n$. Eigenvalues of both matrices are indexed in descending order.

B.1. Perturbation in Eigenvalues and Eigenvectors

The following two lemmas bound the changes in eigenvalues, eigenvectos and the alignment between labels and eigenvectors caused by ξ.

Lemma B.2. *We have the following bound for eigenvalues of $\widetilde{A}$:*

$$\begin{cases} 1 - O(K^{5/2}\xi) \leq \widetilde{\lambda}_i \leq 1, & i = 1, 2, \ldots, \bar{K} \\ \widetilde{\lambda}_i \leq O(\sqrt{\delta} + \bar{K}^{5/2}\xi), & i = \bar{K}+1, \ldots, n \end{cases}$$

Proof. From lemma A.4 we know that

$$\begin{cases} \lambda_i = 1, & i = 1, 2, \ldots, \bar{K}, \\ \lambda_i \leq O(\sqrt{\delta}), & i = \bar{K}+1, \ldots, n. \end{cases} \tag{21}$$

By Weyl's inequality on perturbation, we have

$$|\widetilde{\lambda}_i - \lambda_i| \leq \|E\|_2.$$

The right-hand-side is $\leq \|E\|_F$ which is $O(\bar{K}^{5/2}\xi)$ by lemma B.1. Combining the preceding with equation 21 completes the proof.

□

Lemma B.3. *The norm of the projection of $\boldsymbol{Y}$ onto the range of $\widetilde{V}_I$ is bounded from below, i.e.,*

$$\|\widetilde{V}_I\widetilde{V}_I^T\boldsymbol{Y}\|_F^2 \geq \|V_IV_I^T\boldsymbol{Y}\|_F^2 - O(n\bar{K}^2\xi).$$

Proof. By lemma A.4 we have

$$\lambda_{\bar{K}} - \lambda_{\bar{K}+1} \geq 1 - O(\sqrt{\delta}).$$

By Wedin's Theorem (Wedin, 1972; Stewart, 1990), we have the following bound on the principle angle between the range of $\widetilde{V}_I$ and the range of V_I if $1 \geq O(\sqrt{\delta} + \bar{K}^{5/2}\xi)$

$$\|V_IV_I^T(\widetilde{V}_I\widetilde{V}_I^T - I)\|_F \leq O\left(\frac{\|E\|_F}{\lambda_{\bar{K}} - \lambda_{\bar{K}+1}}\right) \leq O\left(\frac{O(\bar{K}^{5/2}\xi)}{1 - O(\sqrt{\delta})}\right) = O(\bar{K}^{5/2}\xi).$$

Thus

$$\begin{aligned}
\|\widetilde{V}_I\widetilde{V}_I^T\boldsymbol{Y}\|_F^2 =& \|\widetilde{V}_I\widetilde{V}_I^T\boldsymbol{Y} - V_IV_I^T\boldsymbol{Y} + V_IV_I^T\boldsymbol{Y}\|_F^2 \\
\geq& \|V_IV_I^T\boldsymbol{Y}\|_F^2 + 2\langle \widetilde{V}_I\widetilde{V}_I^T\boldsymbol{Y} - V_IV_I^T\boldsymbol{Y},\ V_IV_I^T\boldsymbol{Y}\rangle_F \\
\geq& \|V_IV_I^T\boldsymbol{Y}\|_F^2 + 2\operatorname{Tr}((\widetilde{V}_I\widetilde{V}_I^T - V_IV_I^T)\boldsymbol{Y}\boldsymbol{Y}^TV_IV_I^T) \\
=& \|V_IV_I^T\boldsymbol{Y}\|_F^2 + 2\operatorname{Tr}(V_IV_I^T(\widetilde{V}_I\widetilde{V}_I^T - I)\boldsymbol{Y}\boldsymbol{Y}^T) \\
=& \|V_IV_I^T\boldsymbol{Y}\|_F^2 + 2\langle V_IV_I^T(\widetilde{V}_I\widetilde{V}_I^T - I),\ \boldsymbol{Y}\boldsymbol{Y}^T\rangle_F \\
\geq& \|V_IV_I^T\boldsymbol{Y}\|_F^2 - 2\|V_IV_I^T(\widetilde{V}_I\widetilde{V}_I^T - I)\|_F\|\boldsymbol{Y}\boldsymbol{Y}^T\|_F \\
\geq& \|V_IV_I^T\boldsymbol{Y}\|_F^2 - O(\bar{K}^{5/2}\xi)\frac{n}{\sqrt{K}} \\
=& \|V_IV_I^T\boldsymbol{Y}\|_F^2 - O(n\bar{K}^2\xi).
\end{aligned}$$

□

	$\|\Pi_{\mathcal{I}}(\boldsymbol{y})\|_F$	$\|\Pi_{\mathcal{N}}(\boldsymbol{y})\|_F$	$\|\boldsymbol{J}\boldsymbol{J}^T\boldsymbol{y}\|_F/\|\boldsymbol{J}\boldsymbol{J}^T\|_F$
Pretrained for 1000 epochs	10.063	29.979	3.184
Pretrained for 100 epochs	10.036	29.988	3.175
Randomly initialized	10.014	29.995	3.055

Table 2. Alignment between the Jacobian matrix and the clean labels.

B.2. Error under Gaussian Noise (Proof for Theorem 3.3)

Considering ξ, equation 10 can be rewritten as

$$\begin{aligned}\sum_{i=1}^{p}(\widetilde{\beta}_i^2-2\widetilde{\beta}_i)\sum_{j=1}^{\bar{K}}(\widetilde{v}_i^\top y_j)^2 \leq& \sum_{\bar{k}=1}^{\bar{K}}(\widetilde{\beta}_{\bar{k}}^2-2\widetilde{\beta}_{\bar{k}})\sum_{j=1}^{\bar{K}}(\widetilde{v}_{\bar{k}}^\top y_j)^2\\ \leq& -(1-(1-\eta\widetilde{\lambda}_{\bar{K}})^{2t})\sum_{\bar{k}=1}^{\bar{K}}\sum_{j=1}^{K}(\widetilde{v}_{\bar{k}}^\top y_j)^2\\ =& -(1-(1-\eta\widetilde{\lambda}_{\bar{K}})^{2t})\|\widetilde{V}_I\widetilde{V}_I^T\boldsymbol{Y}\|_F^2,\end{aligned} \tag{22}$$

where $\widetilde{\beta}_i = 1-(1-\eta\widetilde{\lambda}_i)^t$. By lemma A.2 we get the lower bound for $\|V_I V_I^T\boldsymbol{Y}\|_F^2$

$$\begin{aligned}\|V_I V_I^T\boldsymbol{Y}\|_F^2 =& \sum_{\bar{k}}^{\bar{K}}\|v_{\bar{k},1}\|_1^2\\ \geq& n-(n-\bar{K})O(\delta).\end{aligned} \tag{23}$$

Combining lemma B.2, lemma B.3, equation 22 and equation 23 yields the bound for the bias

$$\text{bias} \leq (n-\bar{K})O(\delta)+n\bar{K}^2O(\xi)+\frac{n}{2}\left(1-\eta+\eta\bar{K}^{5/2}O(\xi)\right)^{2t}$$

To bound the variance we first rewrite equation 15 as

$$\begin{aligned}\sum_{i=1}^{p}\widetilde{\beta}_i^2 =& \sum_{\bar{k}=1}^{\bar{K}}\widetilde{\beta}_{\bar{k}}^2+\sum_{\bar{k}=\bar{K}+1}^{p}\widetilde{\beta}_{\bar{k}}^2\\ \leq& \bar{K}(1-(1-\eta)^t)^2+\sum_{\bar{k}=\bar{K}+1}^{p}\widetilde{\beta}_{\bar{k}}^2\\ \leq& \bar{K}(1-(1-\eta)^t)^2+(p-\bar{K})\widetilde{\beta}_{\bar{K}+1}^2.\end{aligned} \tag{24}$$

Combining lemma B.2 and lemma A.4 yields the bound for the variance

$$\text{variance} \leq \frac{\sigma^2}{2}\bar{K}\left(1-(1-\eta)^t\right)^2+\frac{\sigma^2}{2}(p-\bar{K})\left(1-\left(1-\eta O(\sqrt{\delta})-\eta\bar{K}^{5/2}O(\xi)\right)^t\right)^2.$$

C. Contrastive Learning Slightly Improves the Alignment Between Jacobian Matrix and True Labels

We compare the alignments between the clean label vector and the initial Jacobian matrix of (1) network pretrained using SimCLR for 1000 epochs, (2) network pretrained using SimCLR for 100 epochs and (3) randomly initialized network. $\boldsymbol{y}\in\mathbb{R}^{nK}$ is the vector obtained by flattening the label matrix $\boldsymbol{Y}$, i.e., concatenating together all the n rows of $\boldsymbol{Y}$. Let $z(x_i,\boldsymbol{W})\in\boldsymbol{R}^K$ be the output of the network given example x_i and parameters $\boldsymbol{W}\in\mathbb{R}^d$ (we see the parameters of the network as a vector). Then the Jacobian $\boldsymbol{J}$ is defined as

$$\boldsymbol{J}(\boldsymbol{W})=\left[\frac{\partial z(x_1,\boldsymbol{W})}{\boldsymbol{W}}\cdots\frac{\partial z(x_n,\boldsymbol{W})}{\boldsymbol{W}}\right]^\top.$$

Note that $\frac{\partial z(x_i, \boldsymbol{W})}{\boldsymbol{W}} \in \mathbb{R}^{d \times K}$, therefore $\boldsymbol{J}(\boldsymbol{W}) \in \mathbb{R}^{nK \times d}$. In table 2 $\Pi_I(\boldsymbol{y})$ is the projection of $\boldsymbol{y}$ onto the span of the 10 singular vectors of $\boldsymbol{J}(\boldsymbol{W}_0)$ with larges singular values and $\Pi_N(\boldsymbol{y})$ is the projection of $\boldsymbol{y}$ onto the span of the remaining singular vectors. Interestingly, pretraining for more epochs leads to larger $\Pi_I(\boldsymbol{y})$ and smaller $\Pi_N(\boldsymbol{y})$ and therefore larger $\|\boldsymbol{J}\boldsymbol{J}^T\boldsymbol{y}\|_F / \|\boldsymbol{J}\boldsymbol{J}^T\|_F$. Whether this slight improvement in the alignment is important or not to the significant improvement in the robustness deserves further investigation.

D. Experiments

We evaluate the effectiveness of contrastive learning in boosting the robustness of deep networks under various levels of label noise. We first consider fine-tuning all layers of a network pre-trained with contrastive learning on noisy labels, and show that it can achieve a comparable generalization performance to the state-of-the-art robust methods. Then, we show that the structure of the representation matrix obtained by contrastive learning can be leveraged by robust methods to achieve a superior generalization performance under extreme noise levels.

For our evaluation, we use artificially corrupted versions of CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009), as well as a subset of the real-world dataset Webvision (Li et al., 2017), which naturally contains noisy labels. Our method was developed using PyTorch (Paszke et al., 2017). We use 1 Nvidia A40 for all experiments.

Baselines. We compare our results with many commonly used baselines for robust training against label noise: (1) F-correction (Patrini et al., 2017) is a two step process, where a neural network is first trained on noisily-labelled data, then retrained using a corrected loss function based on an estimation of the noise transition matrix. (2) Decoupling (Malach & Shalev-Shwartz, 2017) is a meta-algorithm that trains two networks concurrently, only training on examples where the two networks disagree. (3) Co-teaching (Han et al., 2018) also trains two networks simultaneously. Each network selects subsets of clean data with high probability for the other network to train on. (4) MentorNet (Jiang et al., 2018) uses two neural networks, a student and a mentor. The mentor dynamically creates a curriculum based on the student, while the student trains on the curriculum provided by the mentor. (5) D2L (Ma et al., 2018) learns the training data distribution, then dynamically adapts the loss function based on the changes in dimensionality of subspaces during training. (6) INCV (Chen et al., 2019) identifies random subsets of the training data with fewer noisy labels, then applies Co-teaching to iteratively train on subsets found with the most clean labels. (7) T-Revision (Xia et al., 2019) learns the transition matrix efficiently using an algorithm that does not rely on known points with clean labels. (8) L_DMI (Xu et al., 2019) uses a novel information-theoretic loss function based on determinant based mutual information. (9) ELR (Liu et al., 2020) uses semi-supervised learning techniques to regularize based on the early-learning phase of training, to ensure the noisy labels are not overfit. (10) CRUST (Mirzasoleiman et al., 2020) dynamically selects subsets of clean data points by clustering in the gradient space. (11) Mixup (Zhang et al., 2017) smooths the decision boundary by adding linear interpolations of feature vectors and their labels to the dataset.

D.1. Empirical results on artificially corrupted CIFAR

We first evaluate our method on CIFAR-10 and CIFAR-100, which each contain $50,000$ training images, and $10,000$ test images of size $32 \times 32 \times 3$. CIFAR-10 and CIFAR-100 have 10 and 100 classes respectively. We use the same testing protocol as (Xu et al., 2019; Liu et al., 2020; Xia et al., 2019), by evaluating our method on symmetric and asymmetric label noise. For both CIFAR-10 and CIFAR-100, we use symmetric noise ratios of 0.2, 0.5, 0.8, and an asymmetric noise ratio of 0.4.

Effects of Contrastive Learning In our experiments, we first pre-train ResNet-32 (He et al., 2016) using SimCLR (Chen et al., 2020; SimCLR) for 1000 epochs using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 3×10^{-4}, a weight decay of 1×10^{-6} and a batch size of 128. When pre-training, the last linear layer of ResNet-32 is replaced with a 2-layer projection head with an output dimensionality of 64. When pretraining is finished, we replace the projection head with a new, randomly initialized classification layer, and begin training normally. We also report the results when ELR, Mixup, and Crust are applied to fine-tune the pre-trained network. For ELR, we use $\beta = 0.7$ for the temporal ensembling parameter, and $\lambda = 3$ for the regularization strength. For mixup, we use a mixup strength of $\alpha = 1$. For Crust, we choose a coreset ratio of 0.5.

The results are shown in Table 3. We note that SimCLR pretraining leads to an across the board improvement for Crust, ELR, and Mixup. For lower noise ratios, the improvement is marginal. However, for extreme noise ratios, the improvement is more dramatic. In particular, pre-training boosts the performance of Crust by up to 44.1%, ELR by up to 8.2%, and Mixup

Table 3. Average test accuracy (3 runs) on CIFAR-10 and CIFAR-100. The best test accuracy is marked in bold. We note the higher performance of methods that use SimCLR (CL) pretraining, especially under higher noise levels. In particular, under 80% noise, methods see an average of 27.18%, and 15.58% increase in test accuracy for CIFAR-10 and CIFAR-100 respectively. Results marked with (*) are reproduced from publicly available code. E2E refers to end to end fine-tuning the pre-trained network.

Dataset	CIFAR-10				CIFAR-100			
Noise Type	Sym			Asym	Sym			Asym
Noise Ratio	20	50	80	40	20	50	80	40
F-correction	85.1 ± 0.4	76.0 ± 0.2	34.8 ± 4.5	83.6 ± 2.2	55.8 ± 0.5	43.3 ± 0.7	–	42.3 ± 0.7
Decoupling	86.7 ± 0.3	79.3 ± 0.6	36.9 ± 4.6	75.3 ± 0.8	57.6 ± 0.5	45.7 ± 0.4	–	43.1 ± 0.4
Co-teaching	89.1 ± 0.3	82.1 ± 0.6	16.2 ± 3.2	84.6 ± 2.8	64.0 ± 0.3	52.3 ± 0.4	–	47.7 ± 1.2
MentorNet	88.4 ± 0.5	77.1 ± 0.4	28.9 ± 2.3	77.3 ± 0.8	63.0 ± 0.4	46.4 ± 0.4	–	42.4 ± 0.5
D2L	86.1 ± 0.4	67.4 ± 3.6	10.0 ± 0.1	85.6 ± 1.2	12.5 ± 4.2	5.6 ± 5.4	–	14.1 ± 5.8
INCV	89.7 ± 0.2	84.8 ± 0.3	52.3 ± 3.5	86.0 ± 0.5	60.2 ± 0.2	53.1 ± 0.4	–	50.7 ± 0.2
T-Revision	79.3 ± 0.5	78.5 ± 0.6	36.2 ± 1.6	76.3 ± 0.8	52.4 ± 0.3	37.6 ± 0.3	–	32.3 ± 0.4
L_DMI	84.3 ± 0.4	78.8 ± 0.5	20.9 ± 2.2	84.8 ± 0.7	56.8 ± 0.4	42.2 ± 0.5	–	39.5 ± 0.4
Crust*	85.3 ± 0.5	86.8 ± 0.3	33.8 ± 1.3	76.7 ± 3.4	62.9 ± 0.3	55.5 ± 1.1	18.5 ± 0.8	52.5 ± 0.4
Mixup	89.7 ± 0.7	84.5 ± 0.3	40.7 ± 1.1	86.3 ± 0.1	64.0 ± 0.4	53.4 ± 0.5	15.1 ± 0.1	54.4 ± 2.0
ELR*	90.6 ± 0.6	87.7 ± 1.0	69.5 ± 5.0	86.6 ± 2.9	63.6 ± 1.7	52.5 ± 4.2	23.4 ± 1.9	59.7 ± 0.1
CL+E2E*	88.8 ± 0.5	82.8 ± 0.2	72.0 ± 0.3	83.5 ± 0.5	63.5 ± 0.2	56.1 ± 0.3	$\mathbf{36.7 \pm 0.3}$	52.4 ± 0.2
CL+Crust*	86.5 ± 0.7	87.6 ± 0.3	$\mathbf{77.9 \pm 0.3}$	85.9 ± 0.4	63.0 ± 0.8	$\mathbf{58.3 \pm 0.1}$	34.8 ± 1.5	53.3 ± 0.7
CL+Mixup*	90.8 ± 0.2	84.6 ± 0.4	74.8 ± 0.3	87.5 ± 1.3	64.4 ± 0.4	55.5 ± 0.1	30.3 ± 0.4	55.5 ± 0.8
CL+ELR*	$\mathbf{91.3 \pm 0.0}$	$\mathbf{89.1 \pm 0.1}$	77.7 ± 0.2	$\mathbf{89.7 \pm 0.3}$	$\mathbf{64.7 \pm 0.2}$	55.6 ± 0.2	35.9 ± 0.3	$\mathbf{63.6 \pm 0.1}$

by up to 34.1% under 80% noise. We also note that under 80% noise, SimCLR pretraining alone outperforms all methods without SimCLR pretraining on CIFAR-10 and CIFAR-100.

D.2. Empirical Study on WebVision

WebVision is large scale image dataset with noisy labels (Li et al., 2017). It contains 2.4 million images crawled from Google Images search and Flickr that share the same 1000 classes as the ImageNet dataset. The noise ratio in classes varies from 0.5% to 88%, and the number of images per class varies from 300 to more than 10,000 (Fig. 4 in (Li et al., 2017) shows the noise distribution). We follow the setting in (Jiang et al., 2018) and create a mini WebVision dataset that consists of the top 50 classes in the Google subset with 66,000 images. We use both WebVision and ImageNet test sets for testing the performance of the model. We train InceptionResNet-v2 (Szegedy et al., 2017) for 120 epochs with a starting learning rate of 0.02, which we anneal by a factor of 0.01 at epochs 40 and 80. We use the SGD optimizer with a weight decay of 1×10^{-3}, and a minibatch size of 32. For Mixup and ELR, we use the same hyperparameters as CIFAR.

Table 1 shows the Top-1 and Top-5 accuracy of different methods evaluated on WebVision and ImageNet. We see that for both ELR and Mixup, SimCLR pretraining leads to average improvements of 4.11% and 3.20% for mini Webvision and ImageNet respectively. Furthermore, we note that SimCLR pretraining on its own outperforms every method without SimCLR pretraining, except ELR and Crust.

D.3. Training Only the Linear Layer v.s. Training All Layers

Figure 2 compares the performance of fine-tuning only the last linear layer (i.e., with the encoder frozen) and fine-tuning all layers (i.e., with the encoder unfrozen). For both CIFAR-10 and CIFAR-100 we first pretrain Res-Net 32 using SimCLR for 1000 epochs using the Adam optimizer with a learning rate of 3×10^{-4}, a weight decay of 1×10^{-6} and a batch size of 128. Then, for both frozen and unfrozen fine-tunings, we use the SGD optimizer with a learning rate of 5×10^{-3}, a weight decay of 1×10^{-3}, a batch size of 64. Interestingly, in most cases training all layers achieves a higher test accuracy, which, however, does not mean it is necessarily better under some other hyperparameter settings. Also, we note that training all layers is more likely to overfit, especially under large noise level (column 3 in figure 2).

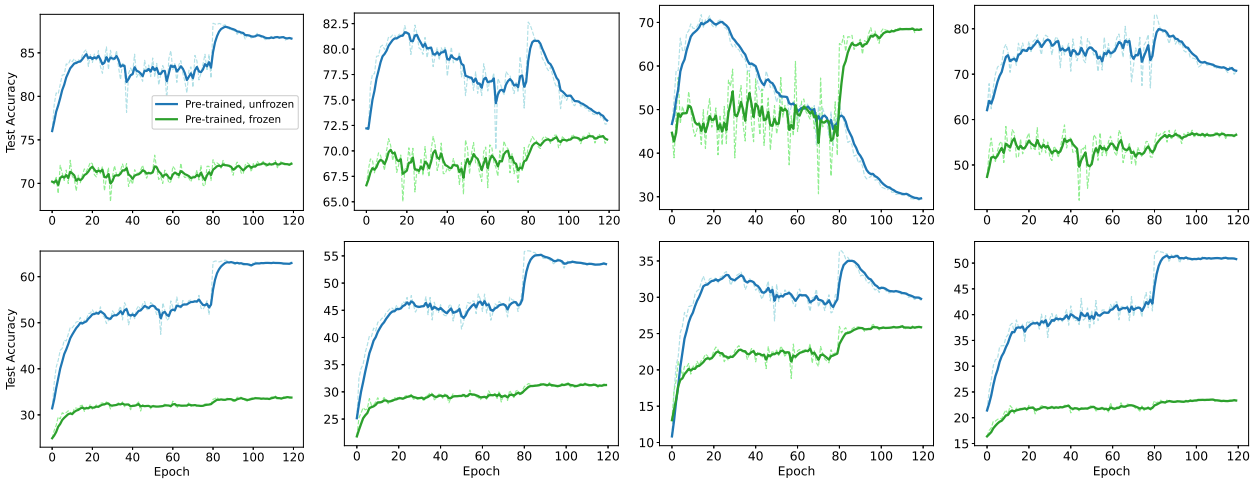


Figure 2. Test accuracy of fine-tuning a pre-trained network with frozen encoder v.s. unfrozen encoder on CIFAR-10 (top) and CIFAR-100 (bottom) under 20%, 50%, 80% symmetric noise and 40% asymmetric noise (left to right).