

---

# Self-Damaging Contrastive Learning

---

Ziyu Jiang<sup>1</sup> Tianlong Chen<sup>2</sup> Bobak Mortazavi<sup>1</sup> Zhangyang Wang<sup>2</sup>

## Abstract

The recent breakthrough achieved by contrastive learning accelerates the pace for deploying unsupervised training on real-world data applications. However, unlabeled data in reality is commonly imbalanced and shows a long-tail distribution, and it is unclear how robustly the latest contrastive learning methods could perform in the practical scenario. This paper proposes to explicitly tackle this challenge, via a principled framework called *Self-Damaging Contrastive Learning (SDCLR)*, to automatically balance the representation learning without knowing the classes. Our *main inspiration* is drawn from the recent finding that deep models have difficult-to-memorize samples, and those may be exposed through network pruning (Hooker et al., 2020). It is further natural to hypothesize that long-tail samples are also tougher for the model to learn well due to insufficient examples. Hence, the *key innovation* in SDCLR is to create a dynamic *self-competitor* model to contrast with the target model, which is a pruned version of the latter. During training, contrasting the two models will lead to adaptive online mining of the most easily forgotten samples for the current target model, and implicitly emphasize them more in the contrastive loss. Extensive experiments across multiple datasets and imbalance settings show that SDCLR significantly improves not only overall accuracies but also balancedness, in terms of linear evaluation on the full-shot and few-shot settings. Our code is available at <https://github.com/VITA-Group/SDCLR>.

## 1. Introduction

### 1.1. Background and Research Gaps

Contrastive learning (Chen et al., 2020a; He et al., 2020; Grill et al., 2020; Jiang et al., 2020; You et al., 2020) recently prevails for deep neural networks (DNNs) to learn

powerful visual representations from unlabeled data. The state-of-the-art contrastive learning frameworks consistently benefit from using bigger models and training on more task-agnostic unlabeled data (Chen et al., 2020b). The predominant promise implied by those successes is to leverage contrastive learning techniques to pre-train strong and transferable representations from internet-scale sources of unlabeled data. However, going from the controlled benchmark data to uncontrolled real-world data will run into several gaps. For example, most natural image and language data exhibit a Zipf long-tail distribution where various feature attributes have very different occurrence frequencies (Zhu et al., 2014; Feldman, 2020). Broadly speaking, such imbalance is not only limited to the standard single-label classification with majority versus minority class (Liu et al., 2019), but also can extend to multi-label problems along many attribute dimensions (Sarafianos et al., 2018). That naturally questions whether contrastive learning can still generalize well in those long-tail scenarios.

We are *not* the first to ask this important question. Earlier works (Yang & Xu, 2020; Kang et al., 2021) pointed out that when the data is imbalanced by class, contrastive learning can learn more balanced feature space than its supervised counterpart. Despite those preliminary successes, we find that the state-of-the-art contrastive learning methods remain certain vulnerability to the long-tailed data (even indeed improving over vanilla supervised learning), after digging into more experiments and imbalance settings (see Sec 4). Such vulnerability is reflected on the linear separability of pre-trained features (the instance-rich classes has much more separable features than instance-scarce classes), and affects downstream tuning or transfer performance. To conquer this challenge further, the main hurdle lies in the absence of class information; therefore, existing approaches for supervised learning, such as re-sampling the data distribution (Shen et al., 2016; Mahajan et al., 2018) or re-balancing the loss for each class (Khan et al., 2017; Cui et al., 2019; Cao et al., 2019), cannot be straightforwardly made to work here.

### 1.2. Rationale and Contributions

Our overall goal is to find a bold push to extend the loss re-balancing and cost-sensitive learning ideas (Khan et al., 2017; Cui et al., 2019; Cao et al., 2019) into an unsupervised setting. The **initial hypothesis** arises from the recent

---

<sup>1</sup>Texas A&M University <sup>2</sup>University of Texas at Austin. Correspondence to: Zhangyang Wang <atlaswang@utexas.edu>.

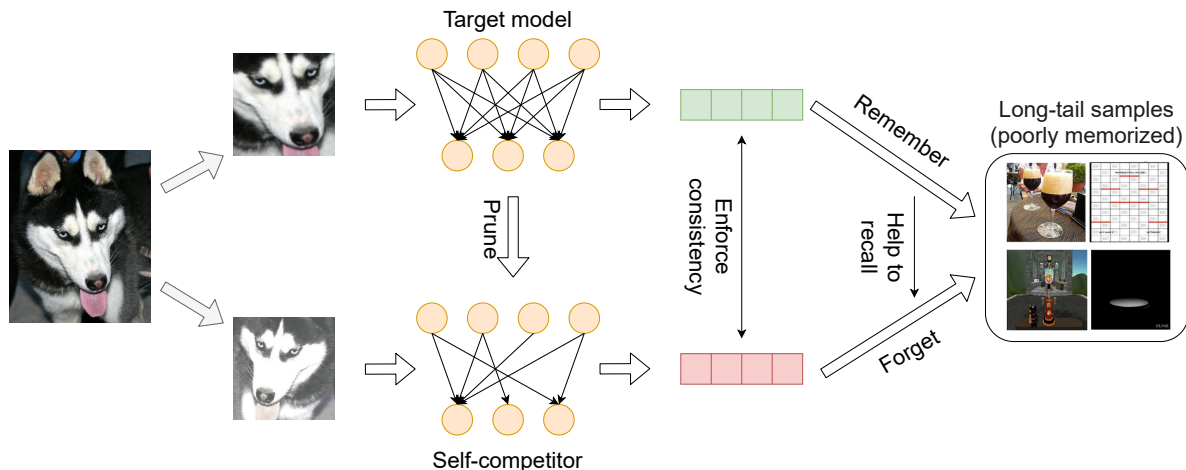


Figure 1. The overview of the proposed SDCLR framework. Built on top of simCLR pipeline (Chen et al., 2020a) by default, the uniqueness of SDCLR lies in its two different network branches: one is the target model to be trained, and the other “self-competitor” model that is pruned from the former online. The two branches share weights for their non-pruned parameters. Either branch has its independent batch normalization layers. Since the self-competitor is always obtained and updated from the latest target model, the two branches will co-evolve during training. Their contrasting will implicitly give more weights on long-tail samples.

observations that DNNs tend to prioritize learning simple patterns (Zhang et al., 2016; Arpit et al., 2017; Liu et al., 2020; Yao et al., 2020; Han et al., 2020; Xia et al., 2021). More precisely, the DNN optimization is content-aware, taking advantage of patterns shared by more training examples, and therefore inclined towards memorizing the majority samples. Since long-tail samples are underrepresented in the training set, they will tend to be poorly memorized, or more “easily forgotten” by the model - a characteristic that one can potentially leverage to spot long-tail samples from unlabeled data in a model-aware yet class-agnostic way.

However, it is in general tedious, if ever feasible, to measure how well each individual training sample is memorized in a given DNN (Carlini et al., 2019). One **blessing** comes from the recent empirical finding (Hooker et al., 2020) in the context of image classification. The authors observed that, network pruning, which usually removes the smallest-magnitude weights in a trained DNN, does not affect all learned classes or samples equally. Rather, it tends to disproportionately hamper the DNN memorization and generalization on the long-tailed and most difficult images from the training set. In other words, long-tail images are not “memorized well” and may be easily “forgotten” by pruning the model, making network pruning a practical tool to spot the samples not yet well learned or represented by the DNN.

Inspired by the aforementioned, we present a principled framework called *Self-Damaging Contrastive Learning* (SDCLR), to automatically balance the representation learning without knowing the classes. The workflow of SDCLR is illustrated in Fig. 1. In addition to creating strong contrastive views by input *data augmentation*, SDCLR intro-

duces another new level of contrasting via “*model augmentation*”, by perturbing the target model’s structure and/or current weights. In particular, the **key innovation** in SDCLR is to create a *dynamic self-competitor* model by pruning the target model online, and contrast the pruned model’s features with the target model’s. Based on the observation (Hooker et al., 2020) that pruning impairs model ability to predict accurately on rare and atypical instances, those samples in practice will also have the largest prediction differences before then pruned and non-pruned models. That effectively boosts their weights in the contrastive loss and leads to **implicit loss re-balancing**. Moreover, since the self-competitor is always obtained from the updated target model, the two models will **co-evolve**, which allows the target model to spot diverse memorization failures at different training stages and to progressively learn more balanced representations. Below we outline our main contributions:

- Seeing that unsupervised contrastive learning is *not immune* to the imbalance data distribution, we design a Self-Damaging Contrastive Learning (SDCLR) framework to address this new challenge.
- SDCLR innovates to leverage the latest advances in understanding DNN memorization. By creating and updating a self-competitor online by pruning the target model during training, SDCLR provides an adaptive online mining process to always focus on the most easily forgotten (long tailed) samples throughout training.
- Extensive experiments across multiple datasets and imbalance settings show that SDCLR can significantly improve not only the balancedness of the learned representation.

## 2. Related works

**Data Imbalance and Self-supervised Learning:** Classical long-tail recognition mainly amplify the impact of tail-class samples, by re-sampling or re-weighting (Cao et al., 2019; Cui et al., 2019; Chawla et al., 2002). However, those hinge on label information and are not directly applicable to unsupervised representation learning. Recently, (Kang et al., 2019; Zhang et al., 2019) demonstrate the learning of feature extractor and classifier head can be decoupled. That suggests the promise of pre-training a feature extractor. Since it is independent of any later task-driven fine-tuning stage, such strategy is compatible with any existing imbalance handling techniques in supervised learning.

Inspired by so, latest works start to explore the benefits of a balanced feature space from self-supervised pre-training for generalization. (Yang & Xu, 2020) presented the first study to utilize self-supervision for overcoming the intrinsic label bias. They observe that simply plugging in self-supervised pre-training, e.g., rotation prediction (Gidaris et al., 2018) or MoCo (He et al., 2020), would outperform their corresponding end-to-end baselines for long tailed classification. Also given more unlabeled data, the labels can be more effectively leveraged in a semi-supervised manner for accurate and debiased classification. reduce label bias in a semi-supervised manner. Another positive result was reported in a (concurrent) piece of work (Kang et al., 2021). The authors pointed out that when the data is imbalanced by class, contrastive learning can learn more balanced feature space than its supervised counterpart.

**Pruning as Compression and Beyond:** DNNs can be compressed of excessive capacity (LeCun et al., 1990) at surprisingly little sacrifice of test set accuracy, and various pruning techniques (Han et al., 2015; Li et al., 2017; Liu et al., 2017) have been popular and effective for that goal. Recently, some works have notably reflected on pruning beyond just an ad-hoc compression tool, exploring its deeper connection with DNN memorization/generalization. (Frankle & Carbin, 2018) showed that there exist highly sparse “critical subnetworks” from the full DNNs, that can be trained in isolation from scratch to reaching the latter’s same performance. That critical subnetwork could be identified by iterative unstructured pruning (Frankle et al., 2019).

The most relevant work to us is (Hooker et al., 2020). For a trained image classifier, pruning it has a non-uniform impact: a fraction of classes, which usually belong to the ambiguous/difficult classes, or the long-tail of less frequent instances. are disproportionately impacted by the introduction of sparsity. That provides novel insights and means to exposing a trained model’s weakness in generalization. For example, (Wang et al., 2021) leveraged this idea to construct an ensemble of self-competitors from one dense model, to troubleshoot an image quality model in the wild.

**Contrasting Different Models:** The high-level idea of SD-CLR, i.e., contrasting two similar competitor models and weighing more on their most disagreed samples, can trace a long history back to the selective sampling framework (Atlas et al., 1990). One most fundamental algorithm is the seminal *Query By Committee* (QBC) (Seung et al., 1992; Gilad-Bachrach et al., 2005). During learning, QBC maintains a space of classifiers that are consistent on predicting all previous labeled samples. At a new unlabeled example, QBC selects two random hypotheses from the space and only queries for the label of the new example if the two disagree. In comparison, our focused problem is in the different realm of unsupervised representation learning.

Spotting two models’ disagreement for troubleshooting either is also an established idea (Popper, 1963). That concept has an interesting link to the popular technique of differential testing (McKeeman, 1998) in software engineering. The idea has also been applied to model comparison and error-spotting in computational vision (Wang & Simoncelli, 2008) and image classification (Wang et al., 2020a). However, none of those methods has considered to construct a self-competitor from a target model. They also work in a supervised active learning setting rather than unsupervised.

Lastly, co-teaching (Han et al., 2018; Yu et al., 2019) performs sample selection in noisy label learning by using two DNNs, each trained on a different subset of examples that have a small training loss for the other network. Its limitation is that the examples that are selected tend to be easier, which may slow down learning (Chang et al., 2017) and hinder generalization to more difficult data (Song et al., 2019). On the opposite, our method is designed to focus on the difficult-to-learn samples in the long tail.

## 3. Method

### 3.1. Preliminaries

**Contrastive Learning.** Contrastive learning learns visual representation via enforcing similarity of the positive pairs  $(v_i, v_i^+)$  and enlarging the distance of negative pairs  $(v_i, v_i^-)$ . Formally, the loss is defined as

$$\mathcal{L}_{\text{CL}} = \frac{1}{N} \sum_{i=1}^N -\log \frac{s(v_i, v_i^+, \tau)}{s(v_i, v_i^+, \tau) + \sum_{v_i^- \in V^-} s(v_i, v_i^-, \tau)} \quad (1)$$

where  $s(v_i, v_i^+, \tau)$  indicates the similarity between positive pairs while  $s(v_i, v_i^-, \tau)$  is the similarity between negative pairs.  $\tau$  represents the temperature hyper-parameter. The negative samples  $v_i^-$  are sampled from negative distribution  $V^-$ . The similarity metric is typically defined as

$$s(v_i, v_i^+, \tau) = \exp(v_i \cdot v_i^+ / \tau) \quad (2)$$

SimCLR (Chen et al., 2020a) is one of the state-of-the-art contrastive learning frameworks. For an input image,

SimCLR would augment it twice with two different augmentations, and then process them with two branches that share the same architecture and weights. Two different versions of the same image are set as positive pairs, and the negative image is sampled from the rest images in the same batch.

**Pruning Identified Exemplars.** (Hooker et al., 2020) systematically investigates the model output changes introduced by pruning and finds that certain examples are particularly sensitive to sparsity. These images most impacted after pruning are termed as *Pruning Identified Exemplars* (PIEs), representing the difficult-to-memorize samples in training. Moreover, the authors also demonstrate that PIEs often show up at the long-tail of a distribution.

We extend (Hooker et al., 2020)’s PIE hypothesis from supervised classification to the unsupervised setting for the first time. Moreover, instead of pruning a trained model and expose its PIEs once, we are now integrating pruning into the training process as an online step. With PIEs dynamically generated by pruning a target model under training, we expect them to expose different long-tail examples during training, as the model continues to be trained. Our experiments show that PIEs answer well to those new challenges.

### 3.2. Self-Damaging Contrastive Learning

**Observation: Contrastive learning is NOT immune to imbalance.** Long-tail distribution fails many supervised approaches build on balanced benchmarks (Kang et al., 2019). Even contrastive learning does not rely on class labels, it still learns the transformation invariances in a data-driven manner, and will be affected by dataset bias (Purushwalkam & Gupta, 2020). Particularly for long-tail data, one would naturally hypothesize that the instance-rich head classes may dominate the invariance learning procedure and leaves the tail classes under-learned.

The concurrent work (Kang et al., 2021) signaled that using the contrastive loss can obtain a balanced representation space that has similar separability (and downstream classification performance) for all the classes, backed by experiments on ImageNet-LT (Liu et al., 2019) and iNaturalist (Van Horn et al., 2018). We independently reproduced and validated their experimental findings. However, we have to point out that it was pre-mature to conclude “contrastive learning is immune to imbalance”.

To see that, we present additional experiments in Section 4.3. While that conclusion might hold for a moderate level of imbalance as presented in current benchmarks, we have constructed a few heavily imbalanced data settings, in which cases contrastive learning will become unable to produce balanced features. In those case, the linear separability of learned representation can differ a lot between head and tail classes. We suggest that our observations complement those in (Yang & Xu, 2020; Kang et al., 2021), that while

(vanilla) contrastive learning can to some extent alleviate the imbalance issue in representation learning, *it does not possess full immunity* and calls for further boosts.

**Our SDCLR Framework.** Figure 1 overviews the high-level workflow of the proposed SDCLR framework. By default, SDCLR is built on top of the simCLR pipeline (Chen et al., 2020a), and follows its most important components such as data augmentations and non-linear projection head. The main difference between simCLR and SDCLR lies in that, simCLR feeds the two augmented images into the same target network backbone (via weight sharing); while SDCLR creates a “self-competitor” by pruning the target model online, and lets the two different branches take the two augmented images to contrast their features.

Specifically, at each iteration we will have a dense branch  $N_1$ , and a sparse branch  $N_2^p$  by pruning  $N_1$ , using the simplest magnitude-based pruning as described in (Han et al., 2015), following the practice of (Hooker et al., 2020). Ideally, the pruning mask of  $N_2^p$  could be updated per iteration after the model weights are updated. In practice, since the backbone is a large DNN and its weights will not change much for a single iteration or two, we set the pruning mask to be lazy-updated at the beginning of every epoch, to save computational overheads; all iterations in the same epoch then adopt the same mask<sup>1</sup>. Since the self-competitor is always obtained and updated from the latest target model, the two branches will co-evolve during training.

We sample and apply two different augmentation chains to the input image  $I$ , creating two different versions  $[\hat{I}_1, \hat{I}_2]$ . They are encoded by  $[N_1, N_2^p]$ , and their output features  $[f_1, f_2^p]$  are fed into the nonlinear projection heads to enforce similarity be under the NT-Xent loss (Chen et al., 2020a). Ideally, if the sample is well-memorized by  $N_1$ , pruning  $N_1$  will not “forget” it – thus little extra perturbation will be caused and the contrasting is roughly the same as in the original simCLR. Otherwise, for rare and atypical instances, SDCLR will amplify the prediction differences between then pruned and non-pruned models – hence those samples’ weights be will implicitly increased in the overall loss.

When updating the two branches, note that  $[N_1, N_2^p]$  will share the same weights in the non-pruned part, and  $N_1$  will independently update the remaining part (corresponding to weights pruned to zero in  $N_2^p$ ). Yet, we empirically discover that it helps to let either branch have its independent batch normalization layers, as the features in dense and sparse may show different statistics (Yu et al., 2018).

<sup>1</sup>We also tried to update pruning masks more frequently, and did not find observable performance boosts.

### 3.3. More Discussions on SDCLR

**SDCLR can work with more contrastive learning frameworks.** We focus on implementing SDCLR on top of simCLR for proving the concept. However, our idea is rather plug-and-play and can be applied with almost every other contrastive learning framework adopting the the two-branch design (He et al., 2020; LeCun et al., 1990; Grill et al., 2020). We will explore combining SDCLR idea with them as our immediate future work.

**Pruning is NOT for model efficiency in SDCLR.** To avoid possible confusion, we stress that *we are NOT using pruning for any model efficiency purpose*. In our framework, pruning would be better described as “selective brain damage”. It is mainly used for effectively spotting samples not yet well memorized and learned by the current model. However, as will be shown in Section 4.9, the pruned branch can have a “side bonus”, that sparsity itself can be an effective regularizer that improves few-shot tuning.

**SDCLR benefits beyond standard class imbalance.** We also want to draw awareness that SDCLR can be extended seamlessly beyond the standard single-class label imbalance case. Since SDCLR relies on no label information at all, it is readily applicable to handling various more complicated forms of imbalance in real data, such as the multi-label attribute imbalance (Sarafianos et al., 2018; Yun et al., 2021).

Moreover, even in artificially class-balanced datasets such as ImageNet, there hide more inherent forms of “imbalance”, such as the class-level difficulty variations or instance-level feature distributions (Bilal et al., 2017; Beyer et al., 2020). Our future work would explore SDCLR in those more subtle imbalanced learning scenarios in the real world.

## 4. Experiments

### 4.1. Datasets and Training Settings

Our experiments are based on three popular imbalanced datasets at varying scales: long-tail CIFAR-10, long-tail CIFAR-100 and ImageNet-LT. Besides, to further stretch out contrastive learning’s imbalance handling ability, we also consider a more realistic and more challenging benchmark long-tail ImageNet-100 as well as another long tail ImageNet with a different exponential sampling rule. The long-tail ImageNet-100 contains less classes, which decreases the number of classes that looks similar and thus can be more vulnerable to imbalance.

**Long-tail CIFAR10/CIFAR100:** The original CIFAR-10/ CIFAR-100 datasets consist of 60000  $32\times 32$  images in 10/100 classes. Long tail CIFAR-10/ CIFAR-100 (CIFAR10-LT / CIFAR100-LT) were first introduced in (Cui et al., 2019) by sampling long tail subsets from the original datasets. Its imbalance factor is defined as the class size of the largest class divided by the smallest class. We by

default consider a challenging setting with the imbalance factor set as 100. To alleviate randomness, all experiments are conducted with five different long tail sub-samplings.

**ImageNet-LT:** ImageNet-LT is a widely used benchmark introduced in (Liu et al., 2019). The sample number of each class is determined by a Pareto distribution with the power value  $\alpha = 6$ . The resultant dataset contains 115.8K images, with the sample number per class ranging from 1280 to 5.

**ImageNet-LT-exp:** Another long tail distribution of ImageNet we considered is given by an exponential function (Cui et al., 2019), where the imbalanced factor set as 256 to ensure the the minor class scale is the same as ImageNet-LT. The resultant dataset contains 229.7K images in total.<sup>2</sup>

**Long tail ImageNet-100:** In many fields such as medical, material, and geography, constructing an ImageNet scale dataset is expensive and even impossible. Therefore, it is also worth considering a dataset with a small scale and large resolution. We thus sample a new long tail dataset called ImageNet-100-LT from ImageNet-100 (Tian et al., 2019). The sample number of each class is determined by a down-sampled (from 1000 classes to 100 classes) Pareto distribution used for ImageNet-LT. The dataset contains 12.21K images, with the sample number per class ranging from 1280 to 5<sup>2</sup>.

To evaluate the influence brought by long tail distribution, for each long tail subset, we would sample a balanced subset from the corresponding full dataset with the same total size as the long tail one to disentangle the influences of long tail and sample size.

For all pre-training, we follow the SimCLR recipe (Chen et al., 2020a) including its augmentations, projection head structures. The default pruning ratio is 90% for CIFAR and 30% for ImageNet. We adopt Resnet-18 (He et al., 2016) for small datasets (CIFAR10/CIFAR100), and Resnet-50 for larger datasets (ImageNet-LT/ImageNet-100-LT), respectively. More details on hyperparameters can be found in the supplementary.

### 4.2. How to Measure Representation Balancedness

The balancedness of a feature space can be reflected by the linear separability w.r.t. all classes. To measure the linear separability, we identically follow (Kang et al., 2021) to employ a three-step protocol: *i)* learn the visual representation  $f_v$  on the training dataset with  $\mathcal{L}_{CL}$ . *ii)* training a linear classifier layer  $L$  on the top of  $f_v$  with a labeled balanced dataset (by default, the full dataset where the imbalanced subset is sampled from). *iii)* evaluating the accuracy of the linear classifier  $L$  on the testing set. Hereinafter, we define such

<sup>2</sup>Refer to our code for details of ImageNet-LT-exp and ImageNet-100-LT.

Table 1. Comparing the *linear separability performance* for models learned on balanced subset  $D_b$  and long-tail subset  $D_i$  of CIFAR10 and CIFAR100. *Many*, *Medium* and *Few* are split based on class distribution of the corresponding  $D_i$ .

Dataset	Subset	<i>Many</i>	<i>Medium</i>	<i>Few</i>	All
CIFAR10	$D_b$	$82.93 \pm 2.71$	$81.53 \pm 5.13$	$77.49 \pm 5.09$	$80.88 \pm 0.16$
	$D_i$	$78.18 \pm 4.18$	$76.23 \pm 5.33$	$71.37 \pm 7.07$	$75.55 \pm 0.66$
CIFAR100	$D_b$	$46.83 \pm 2.31$	$46.92 \pm 1.82$	$46.32 \pm 1.22$	$46.69 \pm 0.63$
	$D_i$	$50.10 \pm 1.70$	$47.78 \pm 1.46$	$43.36 \pm 1.64$	$47.11 \pm 0.34$

Table 2. Comparing the *few-shot performance* for models learned on balanced subset  $D_b$  and long-tail subset  $D_i$  of CIFAR10 and CIFAR100. *Many*, *Medium* and *Few* are split according to class distribution of the corresponding  $D_i$ .

Dataset	Subset	<i>Many</i>	<i>Medium</i>	<i>Few</i>	All
CIFAR10	$D_b$	$77.14 \pm 4.64$	$74.25 \pm 6.54$	$71.47 \pm 7.55$	$74.57 \pm 0.65$
	$D_i$	$76.07 \pm 3.88$	$67.97 \pm 5.84$	$54.21 \pm 10.24$	$67.08 \pm 2.15$
CIFAR100	$D_b$	$25.48 \pm 1.74$	$25.16 \pm 3.07$	$24.01 \pm 1.23$	$24.89 \pm 0.99$
	$D_i$	$30.72 \pm 2.01$	$21.93 \pm 2.61$	$15.99 \pm 1.51$	$22.96 \pm 0.43$

accuracy measure as the *linear separability performance*.

To better understand the influence of the balancedness for down-stream tasks, we consider the important practical application of few-shot learning (Chen et al., 2020b). The only difference between measuring few-shot learning performance and measuring *linear separability accuracy* lies in step ii): we use only 1% samples of the full dataset from which the pre-training imbalanced dataset is sampled. Hereinafter, we define the accuracy measure with this protocol as the *few-shot performance*.

We further divide each dataset to three disjoint groups in terms of the size of classes:  $\{Many, Medium, Few\}$ . In subsets of CIFAR10/CIFAR100, *Many* and *Few* each include the largest and smallest  $\frac{1}{3}$  classes, respectively. For instance in CIFAR-100: the classes with [500-106, 105-20, 19-5] samples belong to [Many (34 classes), Medium (33 classes), Few (33 classes)] categories, respectively. In subsets of ImageNet, we follow OLTR (Liu et al., 2019) to define *Many* as classes each with over training 100 samples, *Medium* as classes each with 20-100 training samples and *Few* as classes under 20 training samples. We report the average accuracy for each specified group, and also use the standard deviation (*Std*) among the three groups’ accuracies as another balancedness measure.

### 4.3. Contrastive Learning is NOT Immune to Imbalance

We now investigate if contrastive learning is vulnerable to the long-tail distribution. In this section, we use  $D_i$  to represent the long tail split of a dataset while  $D_b$  denotes its balanced counterpart.

As shown in Table. 1, for both CIFAR10 and CIFAR100,

models pre-trained on  $D_i$  show larger imbalancedness than that on  $D_b$ . For instance, in CIFAR100, while models pre-trained on  $D_b$  show almost the same accuracy for three groups, the accuracy gradually drops from *many* to *few* when pre-training subset switches from  $D_b$  to  $D_i$ . This indicates that the balancedness of contrastive learning is still fragile when trained over the long tail distributions.

We next explore if the imbalanced representation would influence the downstream few-shot learning applications. As shown in Table. 2, in CIFAR10, the *few shot performance* of *Many* drops by 1.07% when switching from  $D_b$  to  $D_i$  while that of *Medium* and *Few* decrease by 5.30% and 6.12%. In CIFAR100, when pre-training with  $D_b$ , the *few-shot performance* on three groups are similar, and it would become imbalanced when the pre-training dataset switches from  $D_b$  to  $D_i$ . In a word, the balancedness of few-shot performance is consistent with the representation balancedness. Moreover, the bias would become even more serious: The gap between *Many* and *Few* enlarge from 6.81% to 21.86% on CIFAR10 and from 6.65% to 14.73% on CIFAR100.

We further study if the imbalance can also influence large scale dataset like ImageNet in Table. 3. For ImageNet-LT and Imagenet-LT-exp, while the imbalancedness of *linear separability performance* shows weak, that problem becomes much more significant for *few-shot performance*. Especially, for Imagenet-LT-exp, the *few-shot performance* of *Many* is 7.96% higher than that of *Few*. The intuition behind this is that the large volume of the balanced fine-tuning dataset could mitigate the influence of imbalancedness from the pre-trained model. When the scale decreases to 100 classes (ImageNet-100), the imbalancedness consistently exists and it can be reflected via both *linear separability performance* and *few-shot performance*.

Table 3. Comparing the *linear separability performance* and *few-shot performance* for models learned on balanced subset  $D_b$  and long-tail subset  $D_i$  of ImageNet and ImageNet-100. We consider two long tail distributions for ImageNet: Pareto and Exp, which corresponds to ImageNet-LT and Imagenet-LT-exp, respectively. *Many*, *Medium* and *Few* are split according to class distribution of the corresponding  $D_i$ .

Dataset	Long tail type	Split type	<i>linear separability</i>				<i>few-shot</i>			
			<i>Many</i>	<i>Medium</i>	<i>Few</i>	All	<i>Many</i>	<i>Medium</i>	<i>Few</i>	All
ImageNet	Pareto	$D_b$	58.03	56.02	56.71	56.89	29.26	26.97	27.82	27.97
		$D_i$	58.56	55.71	56.66	56.93	31.36	26.21	27.21	28.33
ImageNet	Exp	$D_b$	57.46	57.70	57.02	57.42	32.31	32.91	32.17	32.45
		$D_i$	58.37	56.97	56.27	57.43	35.98	29.56	28.02	32.12
ImageNet-100	Pareto	$D_b$	68.87	66.33	61.85	66.74	48.82	44.71	41.08	45.84
		$D_i$	69.54	63.71	59.69	65.46	48.36	39.00	35.23	42.16

Table 4. Compare the proposed SDCLR with SimCLR in terms of the *linear separability performance*.  $\uparrow$  means the metric the higher the better and  $\downarrow$  means the metric is the lower the better.

Dataset	Framework	<i>Many</i> $\uparrow$	<i>Medium</i> $\uparrow$	<i>Few</i> $\uparrow$	<i>Std</i> $\downarrow$	All $\uparrow$
CIFAR10-LT	SimCLR	78.18 $\pm$ 4.18	76.23 $\pm$ 5.33	71.37 $\pm$ 7.07	5.13 $\pm$ 3.66	75.55 $\pm$ 0.66
	SDCLR	86.44 $\pm$ 3.12	81.84 $\pm$ 4.78	76.23 $\pm$ 6.29	5.06 $\pm$ 3.91	82.00 $\pm$ 0.68
CIFAR100-LT	SimCLR	50.10 $\pm$ 1.70	47.78 $\pm$ 1.46	43.36 $\pm$ 1.64	3.09 $\pm$ 0.85	47.11 $\pm$ 0.34
	SDCLR	58.54 $\pm$ 0.82	55.70 $\pm$ 1.44	52.10 $\pm$ 1.72	2.86 $\pm$ 0.69	55.48 $\pm$ 0.62
ImageNet-100-LT	SimCLR	69.54	63.71	59.69	4.04	65.46
	SDCLR	70.10	65.04	60.92	3.75	66.48

#### 4.4. SDCLR Improves Both Accuracy and Balancedness on Long-tail Distribution

We compare the proposed SDCLR with SimCLR (Chen et al., 2020a) on the datasets that are most easily to be impacted by long tail distribution: CIFAR10-LT, CIFAR100-LT, and ImageNet-100-LT. As shown in Table. 4, the proposed SDCLR leads to a significant *linear separability performance* improvement of **6.45%** in CIFAR10-LT and **8.37%** in CIFAR100-LT. Meanwhile, SDCLR also improve the balancedness by reducing the *Std* by 0.07% in CIFAR10 and 0.23% in CIFAR100. In Imagenet-100-LT, SDCLR achieve an improvement on *linear separability performance* of 1.02% while reducing the *Std* by 0.29.

On few-shot settings, as shown in Table. 5, the proposed SDCLR consistently improves the *few-shot performance* by [3.39%, 2.31%, 0.22%] while decreasing the *Std* by [2.81, 2.29, 0.45] in [CIFAR10, CIFAR100, Imagenet-100-LT], respectively.

#### 4.5. SDCLR Helps Downstream Long Tail Tasks

SDCLR is a pre-training approach that is fully compatible with almost any existing long-tail algorithm. To show that, on CIFAR-100-LT with the imbalance factor of 100, we use SDCLR as pre-training, to fine-tune a SOTA long-tail algorithm RIDE (Wang et al., 2020b) on its top. With SD-

CLR pre-training, the overall accuracy can reach 50.56%, super-passing the original RIDE result by 1.46%. Using SimCLR pre-training for RIDE only spots 50.01% accuracy.

#### 4.6. SDCLR Improves Accuracy on Balanced Datasets

Even balanced in sample numbers per class, existing datasets can still suffer from more hidden forms of “imbalance”, such as sampling bias, and different classes’ difficulty/ambiguity levels, e.g., see (Bilal et al., 2017; Beyer et al., 2020). To evaluate whether the proposed SDCLR can address such imbalancedness, we further run the proposed framework on balanced datasets: The full dataset of CIFAR10 and CIFAR100. We compare SDCLR with SimCLR following standard linear evaluation protocol (Chen et al., 2020a) (On the same dataset, it first pre-trains the backbone, and then finetunes one linear layer on the top of the output features).

The results are shown in Table 6. Note here the *Std* denotes the standard deviation of classes as we are studying the imbalance caused by the varying difficulty of classes. The proposed SDCLR can boost the linear evaluation accuracy by [0.39%, 3.48%] while reducing the *Std* by [1.0, 0.16] in [CIFAR10, CIFAR100], respectively, proving that the proposed method can also help to improve the balancedness even in the balanced datasets.

Table 5. Compare the proposed SDCLR with SimCLR in terms of the *few-shot performance*.  $\uparrow$  means the metric the higher the better and  $\downarrow$  means the metric is the lower the better.

Dataset	Framework	Many $\uparrow$	Medium $\uparrow$	Few $\uparrow$	Std $\downarrow$	All $\uparrow$
CIFAR10	SimCLR	76.07 $\pm$ 3.88	67.97 $\pm$ 5.84	54.21 $\pm$ 10.24	9.80 $\pm$ 5.45	67.08 $\pm$ 2.15
	SDCLR	76.57 $\pm$ 4.90	70.01 $\pm$ 7.88	62.79 $\pm$ 7.37	6.99 $\pm$ 5.20	70.47 $\pm$ 1.38
CIFAR100	SimCLR	30.72 $\pm$ 2.01	21.93 $\pm$ 2.61	15.99 $\pm$ 1.51	6.27 $\pm$ 1.20	22.96 $\pm$ 0.43
	SDCLR	29.72 $\pm$ 1.52	25.41 $\pm$ 1.91	20.55 $\pm$ 2.10	3.98 $\pm$ 0.98	25.27 $\pm$ 0.83
Imagenet-100-LT	SimCLR	48.36	39.00	35.23	5.52	42.16
	SDCLR	48.31	39.17	36.46	5.07	42.38

Table 6. Compare the accuracy (%) and Standard deviation (*Std*) among classes in balanced CIFAR10/100.  $\uparrow$  means the metric the higher the better;  $\downarrow$  means the metric is the lower the better.

Datasets	Framework	Accuracy $\uparrow$	Std $\downarrow$
CIFAR10	SimCLR	91.16	6.37
	SDCLR	91.55	5.37
CIFAR100	SimCLR	62.84	14.94
	SDCLR	66.32	14.82

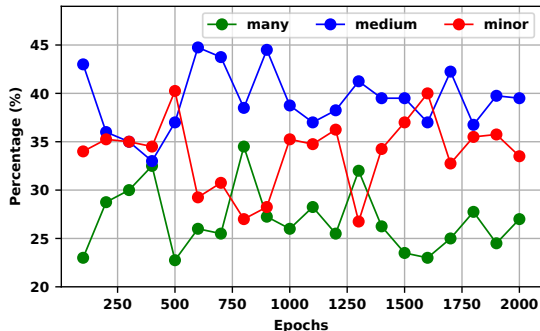


Figure 2. Pre-training on imbalance splits of CIFAR100, The percentage of *many* ( $\bullet$ ), *medium* ( $\bullet$ ) and *few* ( $\bullet$ ) in 1% most easily forgotten data under different training epochs.

#### 4.7. SDCLR Mines More Samples from The Tail

We then measure the distribution of PIEs mined by the proposed SDCLR. Specifically, when pre-training on long tail splits of CIFAR100, we sample top 1% testing data that is most easily influenced by pruning and then evaluate the percentage of *many*, *medium* and *minor* in it under different training epochs. The difficulty of forgetting a sample is defined by the features’ cosine similarity before and after pruning. Figure 2 shows the *minor* and *medium* are much more likely to be impacted comparing to *many*. In particular, while the group distributions of the found PIEs show some variations along with training epochs, in general, we find samples from the *minor* group to gradually increase, while the *many* group samples continue to stay low percentage

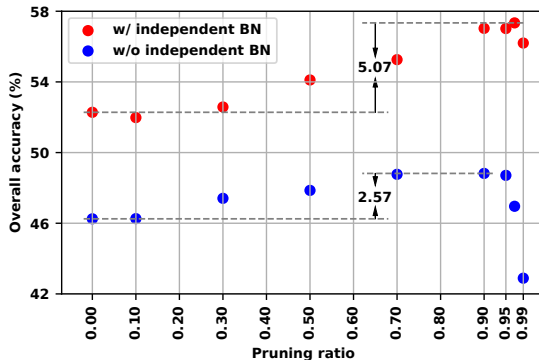


Figure 3. Ablation study of *linear separability performance* w.r.t. the pruning ratios for the dense branch, with ( $\bullet$ ) or without ( $\bullet$ ) independent BNs per branch, on one imbalance split of CIFAR100.

especially when it is close to convergence.

#### 4.8. Sanity Check with More Baselines

**Random dropout baseline:** To verify whether pruning is necessary, we compare with using random dropout (Srivastava et al., 2014) to generate the sparse branch. Under dropout ratio of 0.9, [*linear separability*, *few-shot accuracy*] are [21.99 $\pm$ 0.35%, 15.48 $\pm$ 0.42%], which are much worse than both SimCLR and SDCLR reported in Tab. 4 and 5. In fact, the dropout baseline is often hard to converge.

**Focal loss baseline:** We also compare with the popular focal loss (Lin et al., 2017) for conducting this suggested sanity check. With the best grid searched gamma of 2.0 (grid is [0.5, 1.0, 2.0, 3.0]), it decreases the [accuracy, std] of *linear separability* from [47.33 $\pm$ 0.33%, 2.70 $\pm$ 1.25%] to [46.48 $\pm$ 0.51%, 2.99 $\pm$ 1.01%], respectively. Further analysis shows the contrastive loss scale is not tightly connected with the major or minor class membership as we hypothesized. A possible reason is that the randomness of SimCLR augmentations also notably affects the loss scale.

**Extending to Moco pre-training:** We try MocoV2 (He et al., 2020; Chen et al., 2020c) on CIFAR100-LT. The [accuracy, std] of *linear separability* is [48.23 $\pm$ 0.20%,



$3.50 \pm 0.98\%$ ] and [accuracy,std] of *few-shot performance* is  $[24.68 \pm 0.36\%, 6.67 \pm 1.45\%]$ , respectively, which is worse than SDCLR in Tab 4 and 5.

#### 4.9. Ablation Studies on the Sparse Branch

We study the *linear separability performance* under different pruning ratios in one imbalance subset of CIFAR100. As shown in Figure. 3, the overall accuracies consistently increase with the pruning ratio until it exceeds 90%, which will lead to a quick drop. That shows a trade-off for the sparse branch between being stronger (i.e., needing larger capacity) and being effective in spotting more difficult examples (i.e., needing being sparse).

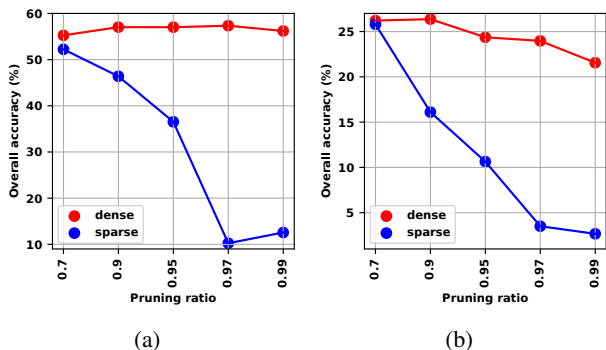


Figure 4. Compare (a) *linear separability performance*, and (b) *few-shot performance*, for representations learned by dense (●) and sparse (●) branches. Both are pre-trained and evaluated on one long tail split of CIFAR100, under different pruning ratios.

We also explore the *linear separability and few-shot performance* of the sparse branch in Figure 4. In the linear separability case (a), the sparse branch quickly lags behind the dense branch when the sparsity goes above 70%, due to limited capacity. Interestingly, even a “weak” sparse branch can still assist the learning of its dense branch. The few shot performance also shows the similar trend.

**The Sparse Branch Architecture** The visualization of pruned ratio for each layer is illustrated in Figure 6. Overall,

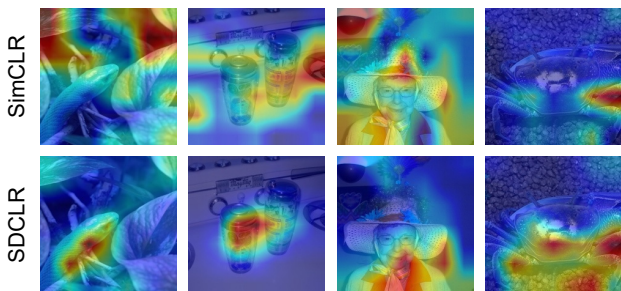


Figure 5. Visualization of attention on tail class images with Grad-CAM (Selvaraju et al., 2017). The first and second row corresponds to SimCLR and SDCLR, respectively.

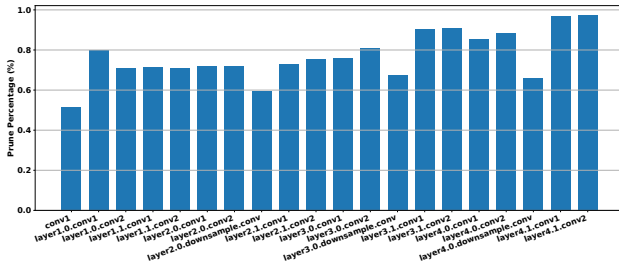


Figure 6. Layer-wise pruning ratio for SDCLR with 90% pruning ratio on Cifar100-LT. The layer following the feed-forward order. We follow (He et al., 2016) for naming each layer.

we find the sparse branch’s deeper layers to be more heavily pruned. This is aligned with the intuition that higher-level features are more class-specific.

**Visualization for SDCLR** We visualize the features of SDCLR and SimCLR on minor classes with Grad-CAM (Selvaraju et al., 2017) in the Figure 5. SDCLR shows to better localize class-discriminative regions for tail samples.

## 5. Conclusion

In this work, we improve the robustness of Contrastive Learning towards imbalance unlabeled data with the principle framework of SDCLR. Our method is motivated the the recent findings that deep models will tend to forget the samples in the long-tail when being pruned. Through extensive experiments across multiple datasets and imbalance settings, we show that SDCLR can significantly mitigate the imbalance. Our future work would explore extending SDCLR to more contrastive learning frameworks.

## References

- Arpit, D., Jastrzkebski, 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 *International Conference on Machine Learning*, pp. 233–242. PMLR, 2017.
- Atlas, L. E., Cohn, D. A., and Ladner, R. E. Training connectionist networks with queries and selective sampling. In *Advances in neural information processing systems*, pp. 566–573. Citeseer, 1990.
- Beyer, L., Hénaff, O. J., Kolesnikov, A., Zhai, X., and Oord, A. v. d. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020.
- Bilal, A., Jourabloo, A., Ye, M., Liu, X., and Ren, L. Do convolutional neural networks learn class hierarchy? *IEEE transactions on visualization and computer graphics*, 24(1):152–162, 2017.

- Cao, K., Wei, C., Gaidon, A., Arechiga, N., and Ma, T. Learning imbalanced datasets with label-distribution-aware margin loss. *arXiv preprint arXiv:1906.07413*, 2019.
- Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., and Song, D. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, pp. 267–284, 2019.
- Chang, H.-S., Learned-Miller, E., and McCallum, A. Active bias: training more accurate neural networks by emphasizing high variance samples. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1003–1013, 2017.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020a.
- Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020b.
- Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020c.
- Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9268–9277, 2019.
- Feldman, V. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 954–959, 2020.
- Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.
- Frankle, J., Dziugaite, G. K., Roy, D. M., and Carbin, M. The lottery ticket hypothesis at scale. *arXiv preprint arXiv:1903.01611*, 8, 2019.
- Gidaris, S., Singh, P., and Komodakis, N. Unsupervised representation learning by predicting image rotations. In *International Conference on Learning Representations*, 2018.
- Gilad-Bachrach, R., Navot, A., and Tishby, N. Query by committee made real. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, pp. 443–450, 2005.
- Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.
- 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. *Advances in Neural Information Processing Systems*, 2018.
- Han, B., Niu, G., Yu, X., Yao, Q., Xu, M., Tsang, I., and Sugiyama, M. Sigua: Forgetting may make learning with noisy labels more robust. In *International Conference on Machine Learning*, pp. 4006–4016. PMLR, 2020.
- Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.
- 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.
- He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.
- Hooker, S., Courville, A., Clark, G., Dauphin, Y., and Frome, A. What do compressed deep neural networks forget?, 2020.
- Jiang, Z., Chen, T., Chen, T., and Wang, Z. Robust pre-training by adversarial contrastive learning. *Advances in Neural Information Processing Systems*, 2020.
- Kang, B., Xie, S., Rohrbach, M., Yan, Z., Gordo, A., Feng, J., and Kalantidis, Y. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019.
- Kang, B., Li, Y., Xie, S., Yuan, Z., and Feng, J. Exploring balanced feature spaces for representation learning. In *International Conference on Learning Representations (ICLR)*, 2021.
- Khan, S. H., Hayat, M., Bennamoun, M., Soheli, F. A., and Togneri, R. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions*

- on neural networks and learning systems, 29(8):3573–3587, 2017.
- LeCun, Y., Denker, J. S., and Solla, S. A. Optimal brain damage. In *Advances in Neural Information Processing Systems*, pp. 598–605, 1990.
- Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient ConvNets. In *International Conference on Learning Representations*, 2017.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.
- Liu, S., Niles-Weed, J., Razavian, N., and Fernandez-Granda, C. Early-learning regularization prevents memorization of noisy labels. *Advances in Neural Information Processing Systems*, 33, 2020.
- Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. Learning efficient convolutional networks through network slimming. In *IEEE International Conference on Computer Vision*, pp. 2736–2744, 2017.
- Liu, Z., Miao, Z., Zhan, X., Wang, J., Gong, B., and Yu, S. X. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2537–2546, 2019.
- Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., and Van Der Maaten, L. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 181–196, 2018.
- McKeeman, W. M. Differential testing for software. *Digital Technical Journal*, 10(1):100–107, 1998.
- 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.
- Popper, K. R. Science as falsification. *Conjectures and refutations*, 1(1963):33–39, 1963.
- Purushwalkam, S. and Gupta, A. Demystifying contrastive self-supervised learning: Invariances, augmentations and dataset biases. *arXiv preprint arXiv:2007.13916*, 2020.
- Sarafianos, N., Xu, X., and Kakadiaris, I. A. Deep imbalanced attribute classification using visual attention aggregation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 680–697, 2018.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.
- Seung, H. S., Oppor, M., and Sompolinsky, H. Query by committee. In *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 287–294, 1992.
- Shen, L., Lin, Z., and Huang, Q. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*, pp. 467–482. Springer, 2016.
- Song, H., Kim, M., and Lee, J.-G. Selfie: Refurbishing unclean samples for robust deep learning. In *International Conference on Machine Learning*, pp. 5907–5915. PMLR, 2019.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.
- Van Horn, G., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., Adam, H., Perona, P., and Belongie, S. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8769–8778, 2018.
- Wang, H., Chen, T., Wang, Z., and Ma, K. I am going mad: Maximum discrepancy competition for comparing classifiers adaptively. In *International Conference on Learning Representations*, 2020a.
- Wang, X., Lian, L., Miao, Z., Liu, Z., and Yu, S. X. Long-tailed recognition by routing diverse distribution-aware experts. *arXiv preprint arXiv:2010.01809*, 2020b.
- Wang, Z. and Simoncelli, E. P. Maximum differentiation (mad) competition: A methodology for comparing computational models of perceptual quantities. *Journal of Vision*, 8(12):8–8, 2008.
- Wang, Z., Wang, H., Chen, T., Wang, Z., and Ma, K. Troubleshooting blind image quality models in the wild. *arXiv preprint arXiv:2105.06747*, 2021.
- Xia, X., Liu, T., Han, B., Gong, C., Wang, N., Ge, Z., and Chang, Y. Robust early-learning: Hindering the memorization of noisy labels. In *International Conference on Learning Representations*, 2021. URL [https://openreview.net/forum?id=Eq15b1\\_hTE4](https://openreview.net/forum?id=Eq15b1_hTE4).
- Yang, Y. and Xu, Z. Rethinking the value of labels for improving class-imbalanced learning. *arXiv preprint arXiv:2006.07529*, 2020.

- Yao, Q., Yang, H., Han, B., Niu, G., and Kwok, J. T.-Y. Searching to exploit memorization effect in learning with noisy labels. In *International Conference on Machine Learning*, pp. 10789–10798. PMLR, 2020.
- You, Y., Chen, T., Sui, Y., Chen, T., Wang, Z., and Shen, Y. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 2020.
- Yu, J., Yang, L., Xu, N., Yang, J., and Huang, T. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018.
- Yu, X., Han, B., Yao, J., Niu, G., Tsang, I., and Sugiyama, M. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pp. 7164–7173. PMLR, 2019.
- Yun, S., Oh, S. J., Heo, B., Han, D., Choe, J., and Chun, S. Re-labeling imagenet: from single to multi-labels, from global to localized labels. *arXiv preprint arXiv:2101.05022*, 2021.
- Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.
- Zhang, J., Liu, L., Wang, P., and Shen, C. To balance or not to balance: An embarrassingly simple approach for learning with long-tailed distributions. *CoRR*, *abs/1912.04486*, 2019.
- Zhu, X., Anguelov, D., and Ramanan, D. Capturing long-tail distributions of object subcategories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 915–922, 2014.

## Supplementary Material: Self-Damaging Contrastive Learning

This supplement contains the following details that we could not include in the main paper due to space restrictions.

- (Sec. 6) Details of the computing infrastructure.
- (Sec. 7) Details of the employed datasets.
- (Sec. 8) Details of the employed hyperparameters.

### 6. Details of computing infrastructure

Our codes are based on Pytorch (Paszke et al., 2017), and all models are trained with GeForce RTX 2080 Ti and NVIDIA Quadro RTX 8000.

### 7. Details of employed datasets

#### 7.1. Downloading link for employed dataset

The datasets we employed are CIFAR10/100, and ImageNet. Their downloading links can be found in Table. 7.

Table 7. Dataset downloading links

Dataset	Link
ImageNet	<a href="http://image-net.org/download">http://image-net.org/download</a>
CIFAR10	<a href="https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz">https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz</a>
CIFAR100	<a href="https://www.cs.toronto.edu/~kriz/cifar-100-python.tar.gz">https://www.cs.toronto.edu/~kriz/cifar-100-python.tar.gz</a>

#### 7.2. Split train, validation and test subset

For CIFAR10, CIFAR100, ImageNet, and ImageNet-100, the testing dataset is set as its official validation datasets. We also randomly select [10000, 20000, 2000] samples from the official training datasets of [CIFAR10/CIFAR100, ImageNet, ImageNet-100] as validation datasets, respectively.

### 8. Details of hyper-parameter settings

#### 8.1. Pre-training

We identically follow SimCLR (Chen et al., 2020a) for pre-training settings except the epochs number. On the full dataset of CIFAR10/CIFAR100, we pre-train for 1000 epochs. In contrast, on sub-sampled CIFAR10/CIFAR100, we would enlarge the pre-training epochs number to 2000 given the dataset size is small. Moreover, the pre-training epochs of ImageNet-LT-exp/ImageNet-100-LT is set as 500.

#### 8.2. Fine-tuning

We employ SGD with momentum 0.9 as the optimizer for all fine-tuning. We follow (Chen et al., 2020c) employing

learning rate of 30 and remove the weight decay for all fine-tuning. When fine-tuning for *linear separability performance*, we train for 30 epochs and decrease the learning rate by 10 times at epochs 10 and 20 as we find more epochs could lead to over-fitting. However, when fine-tuning for *few-shot performance*, we would train for 100 epochs and decrease the learning rate at epoch 40 and 60, given the training set is far smaller.