Robustness to Adversarial Gradients: A Glimpse Into the Loss Landscape of Contrastive Pre-training

Anonymous Authors[1]

Abstract

An in-depth understanding of deep neural network generalization can allow machine learning practitioners to design systems more robust to class balance shift, adversarial attacks, and data drift. However, the reasons for better generalization are not fully understood. Recent works provide empirical arguments suggesting flat minima generalize better. While recently proposed contrastive pre-training methods have also been shown to improve generalization, there is also an incomplete understanding of the loss landscape of these models and why they generalize well. In this work, we analyze the loss landscape of contrastive trained models on the CIFAR-10 dataset by looking at three sharpness measures: (1) the approximate eigenspectrum of the Hessian, (2) $(\mathcal{C}_\epsilon, A)$-sharpness, and (3) robustness to adversarial gradients (RAG), a new efficient measure of sharpness. Our findings suggest models fine-tuned after contrastive training favor flatter solutions relative to baseline classifiers trained with a supervised objective. In addition, our proposed metric yields findings consistent with existing works, demonstrating impacts of learning rate and batch size on minima sharpness.

1. Introduction

Training models that generalize well is a longstanding objective for machine learning scientists. Several works have investigated mechanistic reasons for improved generalization in deep neural networks, arguing that solutions in flat regions generalize better (Keskar et al., 2016; Kaddour et al., 2022; Foret et al., 2020; Jiang* et al., 2020). Recently, researchers have shown the advantages of contrastive learning (CL), a self-supervised learning (SSL) framework, for facilitating improved model performance in low data regimes. Contrastive pre-training methods such as SimCLR (Chen et al., 2020b), MoCo (He et al., 2020), and BYOL (Grill et al., 2020) have demonstrated advantages when only a fraction of the training set is labeled. In addition, the use of contrastive pre-training has lead to improved generalization when fine-tuning a downstream classifier using the CL embeddings.

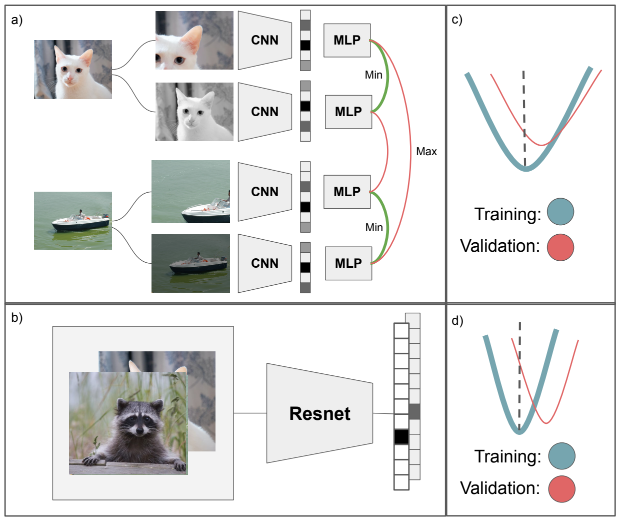

Figure 1. This figure is a visualization of our main hypothesis. Sub-figure (a) represents the workflow for training SimCLR, a SSL approach for learning image representations while (b) is a traditional supervised learning approach. (c) and (d) show hypothetical loss landscapes for SSL and supervised learning respectively. We hypothesize the SSL loss landscape is flatter and less sensitive to shifts in data distribution.

Our primary aim is to better understand the loss landscape of supervised fine-tuning when using representations from pre-trained contrastive SSL algorithms. In this work, we focus our analysis on SimCLR pre-training and observe how the loss landscape curvature of a downstream fine-tuned classifier changes throughout the optimization dynamics. We believe that contrastive pre-training helps push the model weights to solution spaces that favors regions of lower curvature. To this end, we hypothesize that classifiers fine-tuned on pre-trained contrastive SSL objectives will present optimization and loss landscape characteristics of flatter minima

[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.

and lower curvature. Our main contributions are summarized as follows:

1. We train supervised models on CIFAR-10 using standard supervised learning and fine-tuning of pre-trained SimCLR models. We then report performance metrics for both CIFAR-10 test data and on an unseen CIFAR-10.1 (Recht et al., 2018). We find that SimCLR models achieve comparable performance on CIFAR-10 but have smaller generalization gap and improved CIFAR-10.1 performance.

2. We propose a new metric to characterize the generalizability of a solution: *robustness to adversarial gradient* (RAG), based on the idea that a solution that generalizes well should be robust to small changes in the parameters. We show this metric is useful for predicting generalization, highlighting differences in the training dynamics. We validate the applicability of RAG by comparing the metric to existing sharpness measures.

3. We analyze the loss landscape curvature and optimization dynamics by (a) observing changes in loss landscape sharpness during optimization, and (b) computing final sharpness measures using the eigen-spectrum of the approximate Hessian, $(\mathcal{C}_\epsilon, A)$-sharpness, and RAG on the best models. We do this for both supervised models and fine-tuned SimCLR models.

2. Methods for Loss Landscape Analysis

We consider three methods to analyze the loss landscape: (1) approximating the Hessian eigenspectrum using Lanczos algorithm (Lanczos, 1950) (2) $(\mathcal{C}_\epsilon, A)$-sharpness defined by (Keskar et al., 2016), and (3) our proposed robustness to adversarial gradients (RAG) approach. We use all three methods to assess the curvature at converged solution. We also measure RAG throughout training to better understand the learning dynamics of different methods of pre-training.

2.1. Approximate Hessian Spectrum

One approach for determining sharpness is to estimate the top eigenvalues of the Hessian with respect to the parameters. Intuitively, the loss increases more slowly around a flat minima than a sharp minima and thus the top eigenvalues are smaller. However, computing the full Hessian for large models is computationally intractable (Jastrzębski et al., 2018; Foret et al., 2020). As in prior work (Jastrzębski et al., 2018; Ghorbani et al., 2019), we instead approximate the Hessian spectrum via the Lanczos algorithm. A proxy for sharpness used in the literature (e.g. by Jastrzebski et al. (2020)) is the absolute ratio of $|\frac{\lambda_{\max}}{\lambda_5}|$.

Algorithm 1 Computing adversarial gradient

```
Require: Parameters $\phi$, objective function $L$
Require: Step size $\lambda$, dataset $d$
  Cache initial parameters $\phi$
  $r = 0$
  for $i = 1, 2, \ldots, n$ do
    Load parameters $\phi_t \leftarrow \phi$
    Sample minibatch of data $d = (x, y) \sim \mathcal{D}$
    Compute gradient $g = \nabla_\phi \mathcal{L}(f_{\phi_t}(x), y)$
    Do adversarial update $\phi^{adv} \leftarrow \phi_t + \lambda \frac{g}{||g||_2}$
    Sample minibatch of data $d = (x, y) \sim \mathcal{D}$
    $r = r + \mathcal{L}(f_{\phi^{adv}}(x), y)$
  end for
  return $\frac{r}{n}$
```

2.2. $(\mathcal{C}_\epsilon, A)$-sharpness

Although approximating the eigenvalue spectrum of the Hessian yields insight into the curvature of the loss landscape, this approach requires extensive computational overhead in large models. To deal with this constraint, Keskar et al. (2016) propose a more efficient approach of measuring sharpness in a local neighborhood, termed $(\mathcal{C}_\epsilon, A)$-sharpness and defined as

$$\phi_{w,\mathcal{L}}(\epsilon, A) = \frac{\max_{y \in \mathcal{C}_\epsilon}(\mathcal{L}(w + Ay) - \mathcal{L}(w))}{1 + \mathcal{L}(w)} \times 100, \quad (1)$$

where $\mathcal{L}$ is the loss function, w are the model parameters, $A \in \mathbb{R}^{n \times p}$ is a random matrix, and $\mathcal{C}_\epsilon$ is a constrained box around a minima of the loss $\mathcal{L}$.

2.3. Robustness to Adversarial Gradient

We propose and analyze an efficient measure of sharpness: *robustness to adversarial gradient* (RAG). RAG is motivated by the hypothesis that a minima that generalizes well should not be sensitive to small shifts in the input data distribution. Therefore, one adversarial gradient update should not have much impact on the loss. This measure can also be motivated theoretically by Theorem 1 in Foret et al. (2020), which (informally) states:

Theorem 2.1. *For any $\rho > 0$ and any distribution $\mathcal{D}$, with probability $1 - \delta$ over the choice of the training set $S \sim \mathcal{D}$,*

$$\mathcal{L}_\mathcal{D}(w) \leq \max_{\|\epsilon\|_2 \leq \rho} \mathcal{L}_\mathcal{S}(w + \epsilon) + h(\|w\|^2/\rho^2) \quad (2)$$

where h is a strictly increasing function. Hence, approximately maximizing $\mathcal{L}_S$ in a local neighborhood via a gradient ascent step offers a bound on the true generalization performance.

To compute RAG, we sample a random mini-batch and take one gradient ascent step. We evaluate the loss of these new

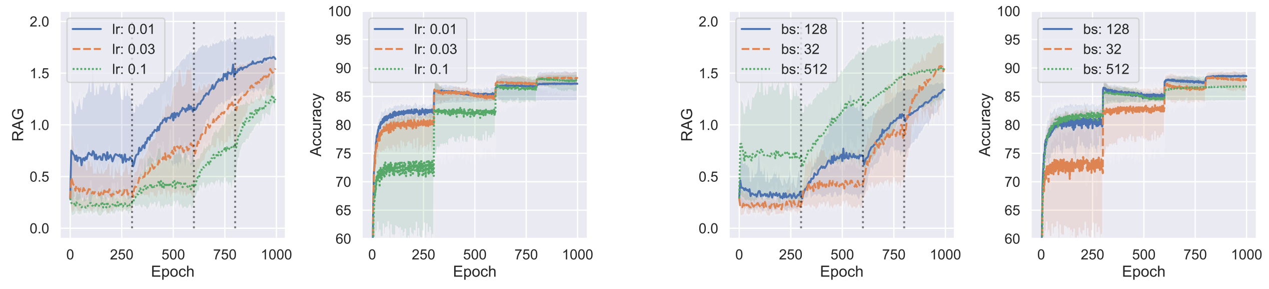

Figure 2. RAG and test accuracy for supervised models throughout training, with varying learning rate (left) and batch size (right). RAG is computed after every 10 epochs of training. Dashed lines represent learning rate decay by a factor of five.

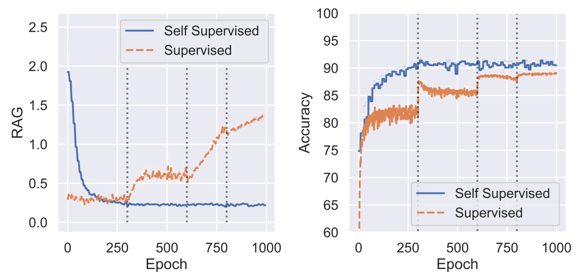

Figure 3. Comparison of RAG and test accuracy for supervised models and models fine-tuned after SimCLR pre-training throughout optimization. RAG is computed after every 10 epochs of training. For SimCLR training, we fine-tune the downstream classifier for 10 epochs before computing RAG on the fine-tuned model. We found 10 epochs achieved sufficient classification accuracy.

parameters on another random mini-batch. We then reset the parameters and repeat this procedure. After n such iterations, we report the average loss on the second random mini-batch. We summarize our approach in Algorithm 1. We define our RAG-sharpness measure as $\frac{\mathcal{L}(w_t^{adv})-\mathcal{L}(w_t)}{1+\mathcal{L}(w_t)}$ to be consistent with equation 1. This formulation makes our sharpness metric contingent on model effectiveness by normalizing on the current loss.

A similar adversarial gradient-based approach was considered by (Iyer et al., 2020) to characterize final minima. Although SAM (Foret et al., 2020) has a similar motivation for their algorithm, to our knowledge, RAG has not been studied explicitly in the context of understanding training dynamics. Due to space constraints, we describe additional details of RAG and related work in Appendix C.5.

3. Results

In this section, we present our results from computing the three measures of sharpness and our analysis of RAG throughout training. To better understand the learning dynamics and how they may impact transfer on downstream tasks, we conduct our experiments on three different approaches to training: (1) standard supervised learning, (2) fine-tuning the final layer of a random feature extractor (random initiliazation of ResNet-18 (Frankle et al., 2020)), and (3) SSL via a contrastive objective (SimCLR), followed by fine-tuning .

We use the same ResNet-18 (He et al., 2016) model to ensure a fair comparison and because residual networks are commonly used in a variety of deep learning applications. We follow standard training procedures for all the models (described in detail in the appendix), except the supervised learning model is trained for 1000 epochs to match what is commonly used in contrastive learning. We present the summary metrics of the best-performing models in Table 1 and the sharpness metrics for different batch sizes in Table 2. We also report results on the CIFAR 10.1 dataset (Recht et al., 2018), which attempted to produce unseen data that follow the CIFAR-10 dataset distributions by following the data collection procedure in the original papers. This is a better proxy for true generalization performance since hyperparameters and architectures have not been implicitly tuned to optimize for such CIFAR 10.1 performance.

In Figure 2, we show how RAG and test accuracy evolve during supervised training for various batch sizes and learning rates. While RAG can scale up to larger models, we found that the sharpness measure by Keskar becomes numerically unstable. In Figure 3, we present how the best-performing SimCLR and supervised model evolve throughout training. SimCLR has consistently lower sharpness and better generalization.

3.1. Discussion

We find that SimCLR results in a flatter training loss landscapes for downstream fine-tuned classifiers, as measured by the top eigenvalues of the Hessian, RAG, and $(\mathcal{C}_\epsilon, A)$-sharpness (Table 2). At the same time, the resulting fine-tuned SimCLR classifiers generalize better (Table 1), especially on CIFAR 10.1. At the points of learning rate decay we observe that RAG sharpness increases; this is consistent with lower learning rate optimization converging to sharper minima. Conversely, we find that SimCLR is robust to the

Model	$\mathcal{L}_{train}$	$\mathcal{L}_{test}$	Test Acc (%)	C-10.1 Acc (%)	$\frac{\mathcal{L}(w_t^{adv})-\mathcal{L}(w_t)}{1+\mathcal{L}(w_t)}$
ResNet18	$\mathbf{0.0007 \pm 0.0000}$	0.5374 ± 0.0234	89.13 ± 0.31	78.12 ± 0.40	1.44 ± 0.06
Random-ResNet18	1.9217 ± 0.0040	1.9181 ± 0.0002	31.33 ± 0.16	24.46 ± 0.74	5.68 ± 0.06
SimClr+ResNet18	0.2538 ± 0.0096	$\mathbf{0.2992 \pm 0.0063}$	$\mathbf{89.75 \pm 0.13}$	$\mathbf{82.05 \pm 0.18}$	$\mathbf{0.08 \pm 0.01}$

Table 1. Performance comparison of classifiers trained using supervised objective and downstream classifiers fine-tuned using representations learned from the self-supervised SimCLR framework. We report the results in this table for the best performing hyperparameter sets in terms of test accuracy. The downstream fine-tuned classifier appears less sensitive to batch size and its effects on generalization relative to the baseline classifier. We fine tune the last layer for 10 epochs for SimCLR. (C-10.1 acc refers to accuracy on CIFAR-10.1 test data).

	Eigenvalue statistics		RAG		$(\mathcal{C}_\epsilon, A)$-sharpness
Model	$\lvert\lambda_{max}\rvert$	$\lvert\frac{\lambda_{max}}{\lambda_5}\rvert$	$\mathcal{L}(w_t^{adv})$	$\frac{\mathcal{L}(w_t^{adv})-\mathcal{L}(w_t)}{1+\mathcal{L}(w_t)}$	$\phi_{w,\mathcal{L}}(\epsilon, A)$
ResNet18$_{512}$	302.3 ± 78.5	1.6 ± 0.4	1.6 ± 0.04	1.6 ± 0.03	131.8 ± 189.6
ResNet18$_{128}$	230.2 ± 41.4	1.5 ± 0.2	1.4 ± 0.03	1.4 ± 0.03	80.8 ± 30.9
Random-ResNet18$_{512}$	1030.5 ± 133.3	1.6 ± 0.2	5.68 ± 0.06	1.29 ± 0.06	463.3 ± 1282.7
Random-ResNet18$_{128}$	1160.3 ± 243.4	1.8 ± 0.3	5.71 ± 0.08	1.25 ± 0.07	53.3 ± 34.6
SimClr+ResNet18$_{512}$	48.7 ± 19.8	7.6 ± 3.5	0.61 ± 0.02	0.21 ± 0.01	1.8 ± 3.8
SimClr+ResNet18$_{128}$	$\mathbf{11.5 \pm 3.8}$	$\mathbf{6.7 \pm 2.7}$	$\mathbf{0.41 \pm 0.02}$	$\mathbf{0.08 \pm 0.01}$	$\mathbf{0.017 \pm 0.002}$

Table 2. Comparison of largest eigenvalues, RAG, and $(\mathcal{C}_\epsilon, A)$-sharpness for models trained via supervised objective only and supervised models trained via representations learned using contrastive objective. $|\lambda_{max}|$ represents the magnitude of the largest eigenvalue and $|\frac{\lambda_{max}}{\lambda_5}|$ is the magnitude of the ratio of the largest eigenvalue and 5th largest eigenvalue, a common proxy for sharpness (Jastrzebski et al., 2020). The model subscript indicates batch size used for training and fine-tuning classifier. All computations are done on train data.

tendency of lower learning rate optimization to settle in sharper valleys (Figure 3).

In Figure 2, we observe that higher learning rates and lower batch sizes yield a lower RAG value throughout the training trajectory. This agrees with prior work that uses other metrics to measure sharpness (Keskar et al., 2016; Jastrzębski et al., 2018). For supervised learning, smaller batch sizes can act as a regularizer through both noisy optimizer updates and regularization with batch normalization. For contrastive learning models, larger batch size can be beneficial for increasing the number of negative samples to avoid mode collapse. This creates a trade-off where larger batch size may cause the model to converge to sharper minima but may also improve the quality of the representation by increasing number of negative samples. While we do not observe the phenomena in our work (see Figure 5), it may occur on larger datasets where it is more difficult to learn a representation.

We observe that the fine-tuning process from the pre-trained SimCLR model results in a reduced generalization gap ($\mathcal{L}_{train} - \mathcal{L}_{test}$) between train and test datasets on CIFAR-10, relative to the supervised learning models. In addition, we see that the fine-tuned downstream classifier yields improved test accuracy on CIFAR-10.1. Although we cannot make a casual association, we hypothesize that this is due to the fine-tuned model finding a flatter solution as supported by RAG and other flatness metrics, demonstrated in Table 2. In follow-up work, we plan to investigate whether generalization performance on related datasets such as SVHN, CIFAR-100, and ImageNet is associated with flatness of model optima.

4. Conclusion

In this work, we seek to understand why CL-based approaches such as SimCLR (Chen et al., 2020b) improve generalization by analyzing properties of the loss landscape and optimization dynamics. Compared to a classifier trained by supervised learning, we find that fine-tuning a pre-trained SSL model, in particular SimCLR, achieves a minima that has smaller Hessian eigenvalues, is more robust to adversarial gradient perturbations, and achieves lower $(\mathcal{C}_\epsilon, A)$-sharpness. We posit this is a reason for its improved generalization. We also study how RAG evolves over time, which highlights a difference in how the solution is found between contrastive learning and supervised learning. We believe a better understanding of the training dynamics will inform more principled algorithm design. Since our metric for sharpness is easy to compute and correlates with generalization, we hope to inspire the discovery of new algorithms and thus achieve better robustness to data distribution shifts and generalization.

References

Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning. *arXiv e-prints*, art. arXiv:2105.04906, May 2021.

Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 699–708, 2020a.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020b.

Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020c.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

Linus Ericsson, Henry Gouk, and Timothy M Hospedales. How well do self-supervised models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5414–5423, 2021.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Jonathan Frankle, David J Schwab, and Ari S Morcos. Training batchnorm and only batchnorm: On the expressive power of random features in cnns. *arXiv preprint arXiv:2003.00152*, 2020.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In *International Conference on Machine Learning*, pages 2232–2241. PMLR, 2019.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised Learning. *arXiv e-prints*, art. arXiv:2006.07733, June 2020.

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

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997.

Nikhil Iyer, V Thejas, Nipun Kwatra, Ramachandran Ramjee, and Muthian Sivathanu. Wide-minima density hypothesis and the explore-exploit learning rate schedule. *arXiv preprint arXiv:2003.03977*, 2020.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Stanisław Jastrzębski, Zachary Kenton, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. On the relation between the sharpest directions of dnn loss and the sgd step length. *arXiv preprint arXiv:1807.05031*, 2018.

Stanislaw Jastrzebski, Maciej Szymczak, Stanislav Fort, Devansh Arpit, Jacek Tabor, Kyunghyun Cho, and Krzysztof Geras. The break-even point on optimization trajectories of deep neural networks. *arXiv preprint arXiv:2002.09572*, 2020.

Yiding Jiang*, Behnam Neyshabur*, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SJgIPJBFvH`.

Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt J Kusner. Questions for flat-minima optimization of modern neural networks. *arXiv preprint arXiv:2202.00661*, 2022.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. 1950.

Amir Gholami Michael Mahoney Joseph Gonzalez Noah Golmant, Zhewei Yao. pytorch-hessian-eigenthings: efficient pytorch hessian eigendecomposition, October 2018. URL `https://github.com/noahgolmant/pytorch-hessian-eigenthings`.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400. PMLR, 2019.

Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. Restricted boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 791–798, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937933. doi: 10.1145/1273496.1273596. URL `https://doi.org/10.1145/1273496.1273596`.

Yuandong Tian, Xinlei Chen, and Surya Ganguli. Understanding self-supervised Learning Dynamics without Contrastive Pairs. *arXiv e-prints*, art. arXiv:2102.06810, February 2021.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding. *arXiv e-prints*, art. arXiv:1807.03748, July 2018.

Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine Learning*, 109(2): 373–440, 2020.

Yifan Zhang, Bryan Hooi, Dapeng Hu, Jian Liang, and Jiashi Feng. Unleashing the power of contrastive self-supervised visual models via contrast-regularized fine-tuning. *Advances in Neural Information Processing Systems*, 34, 2021.

A. Related work

A.1. Loss landscapes and flat minima

In the deep learning literature, there has been various works trying to understand the loss landscape and to discover "flat" minima (Hochreiter and Schmidhuber, 1997). These regions are hypothesized to generalize better as shifts between the train and test distributions keep parameters in the flat, low loss region (Dziugaite and Roy, 2017). One method used to improve generalization to compute an exponential moving average of the weights (Polyak and Juditsky, 1992). This technique has been revisited and demonstrated to work well with modern neural networks in the Stochastic Weight Averaging (SWA) algorithm (Izmailov et al., 2018). SWA averages the weights towards the end of training to try to find a minima in a flatter region. Another technique is to use sharpness aware minimization (SAM) (Foret et al., 2020) which seeks to minimize the worst-case perturbation throughout optimization. Recently, Kaddour et al. (2022) probe at differences between the two approaches and how they are potentially complementary. There are criticisms of the flat minima argument for generalization as one can increase the sharpness of the Hessian via reparameterization (Dinh et al., 2017). However, our minima are found through an optimization procedure rather than by mathematical reparametrization.

A.2. Self-supervised Learning

Machine learning scientists have long sought to utilize the abundance of unlabeled data to improve performance on supervised machine learning tasks (Salakhutdinov et al., 2007; Van Engelen and Hoos, 2020). Over the last few years there has been a resurgence of these unsupervised pre-training techniques, also referred to as self-supervised pre-training. (Chen et al., 2020c) leverage self-supervised pre-training using SimCLRv2 to fine-tune a downstream supervised classifier on ImageNet, beating counterpart state-of-the-art approaches. Others have sought to improve the downstream fine-tuning methodology when using self-supervised pre-trained models (Chen et al., 2020a; Zhang et al., 2021). However, the analysis conducted by (Ericsson et al., 2021), comparing 13 different self-supervised pre-training methods on 40 downstream fine-tuning tasks, demonstrates that there is yet to be dominant method for self-supervised pre-training for all around tasks. Further advancement of self-supervised pre-training and fine-tuning methods may require a deeper understanding of the loss and optimization landscapes.

Recently there has been a particular focus in using contrastive optimization objectives for self-supervised pre-training. In this training procedure a model is pre-trained using a self-supervised loss, such as the InfoNCE loss, which aims to identify the positive sample across a collection of negative samples by optimizing negative log probability of identifying the positive sample x_{t+k}:

$$\mathcal{L}_N = -\mathbb{E}_X[\log \frac{f_k(x_{t+k}, c_t)}{\sum_{x_j \in X} f_k(x_j, c_t)}], \tag{3}$$

where c_t is the context vector, and X is a set of negative samples (van den Oord et al., 2018). SimCLR (Chen et al., 2020b) attempts to map two different augmented views of an image to a similar representation, while maximizing the

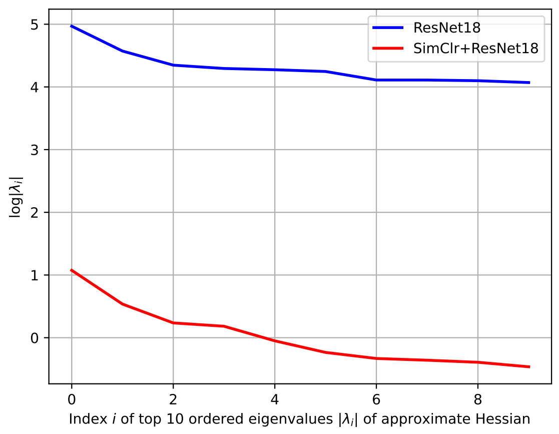

Figure 4. Top 10 log-eigenvalues of approximate Hessian for loss landscape on training data. Supervised model trained using CL representations yields lower magnitudes of eigenvalues, suggesting CL helps models find solutions in lower curvature regions. Magnitude of top eigenvalues for fine-tuned classifier reduce at a quicker rate relative to baseline model.

distance from the representations of other examples. For self-supervised frameworks that do not require negative samples the mode collapse problem is avoided by adding explicit loss preventing a constant prediction. In the case of BYOL this is accomplished through one network being an exponential moving average of the other's weights resulting in distributional differences between two representations (Tian et al., 2021). In contrast, VICReg accomplishes the latter explicitly by containing a loss term that penalizes the degenerate collapsed distributions for each dimension in the learned representation (Bardes et al., 2021).

A.3. Unseen test data

(Recht et al., 2018; 2019) have attempted to produce unseen data that follow the CIFAR-10 and ImageNet dataset distributions by following the data collection procedure in the original papers. They posit there is a danger of machine learning research repeatedly optimizing the same benchmarks and thus overfitting to test sets over time by chasing better benchmark performance. Evaluation on unseen data from the same distribution helps assess this potential problem. Surprisingly, they find more recent models with higher original accuracy show a smaller drop and better overall performance. We use the CIFAR 10.1 dataset (Recht et al., 2018) to assess the benefits of CL approaches and whether metrics associated with sharpness correlate to better generalization performance.

B. Eigen-spectrum of SSL and SL Models

In Figure 2, we show the log-spectrum of the top 10 egienvalues of the approximate Hessian. We see that the solution found by fine-tuning a classifier from the CL representations learned via SimCLR lead to significantly lower eigenvalues, which means there are fewer high curvature directions.

When reporting the proxy-statistic defined by (Jastrzebski et al., 2020) (see Table 2) it is important to consider both the ratio and the value of $\lambda_{\max}$, in which case a high value of of $\lambda_{\max}$ and a ratio close to 1 would imply that there are several dimensions that exhibit high levels of curvature. In contrast, a low value of $\lambda_{\max}$ and a higher ratio would suggest the highest curvature dimension is small and other dimensions exhibit lower degrees of curvature relative to the largest eigenvalue. We hypothesise that a downstream classifier fine-tuned using representations from a contrastive SSL objective will yield lower values for $\lambda_{\max}$ and higher values for $|\frac{\lambda_{\max}}{\lambda_5}|$ relative to a classifier trained under the baseline supervised objective. In effect, this result could suggest that models trained using representations learned by contrastive objectives in SSL models yield flatter minima solutions, and thus also leading to improved generalization.

C. Supplementary Results

C.1. Supervised learning

We train the ResNet-18 architecture using SGD with various learning rates, momentum of 0.9, and weight decay factor of 0.0005 (He et al., 2016). We drop the learning rate by a factor of 5 at the end of the 60th, 120th, and 160th epoch. To assess generalization performance, we use the CIFAR-10.1 dataset (Recht et al., 2018). This dataset acts a proxy for true generalization performance. We present the results on the CIFAR10 test data and CIFAR-10.1 dataset in Table 1. We observe that a higher initial learning rate results in a lower training loss but ends up over-fitting more to the dataset with a higher test loss and worse performance on both the test and unseen test set. For the remainder of the supervised learning experiments, we use a learning rate of 0.1.

C.2. SimClr Self-Supervised Learning

We consider SimClr (Chen et al., 2020b) for CL of representation embeddings on CIFAR-10. We used the Adam optimizer with a fixed learning rate of 0.001 and a batch size of 128 and train for 1000 epochs. We then use the trained embedding encoder of the SimClr architecture with an appended linear layer for the downstream classification task and fine-tune a supervised model using the cross-entropy loss. We do this for batch sizes 128 and 512 for the supervised task and report the final model train loss, test loss, test accuracy, and unseen data (CIFAR-10.1) test accuracy in

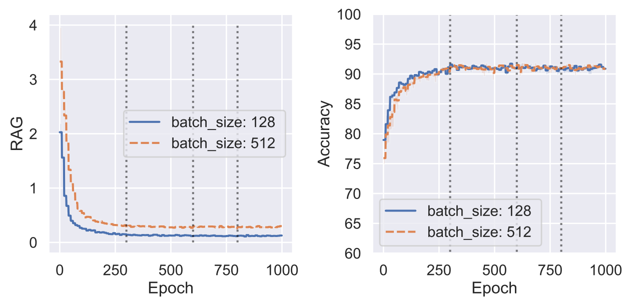

Figure 5. RAG and test accuracy for self-supervised models throughout training, with varying batch size. RAG is computed after every 10 epochs of fine tuning on the classification task. Dashed lines represent learning rate decay by a factor of five

Table 2. We observe that the supervised models trained using the representations learned via SimClr lead to improved generalization on test data and unseen CIFAR-10.1 data relative to the baseline supervised models.

C.3. Loss Landscape Curvature and Sharpness

To compute the approximate Hessian spectrum we use the implementation by Golmant et al (Noah Golmant, 2018). Due to the computational overhead of the method, we compute the eigen-spectrum of the approximate Hessian on mini-batches of data. Consequently, there is variance in the computations. Therefore, we evaluate on 10 random trials and report the mean and standard deviations for $|\lambda_{max}|$ and the ratio $|\frac{\lambda_{max}}{\lambda_5}|$. We present the results of this analysis in Table 3.

We computed RAG values for supervised and self-supervised models with batch sizes 128 and 512 as reported in table 4. For the training set we observe the following three points. First, the supervised models have significantly lower loss on the training data than the self-supervised models. This indicates that these models are over-fit to the training set which is also indicated by model accuracies: 0.98 for supervised and 0.89 for self-supervised. Second, we replicate the results from (Keskar et al., 2016) where small batch supervised learning achieves a smaller ratio in both RAG and magnitude of top Hessian eigenvalues (Noah Golmant, 2018). We do not observe this discrepancy for self-supervised models (more in the discussion). Finally, we find that self-supervised models have significantly lower RAG values than supervised models, indicating their flatter loss landscape.

For the test data sharpness evaluation, we identify that the loss increases for all the models but significantly more for the supervised models. Consistent with that is the decrease in the RAG evaluation, highlighting the connection between the RAG value and how over-fit the model is. We note that for the hessian eigenspectra metric, the largest eigen value increases for the supervised models and stays consistent for self-supervised. Finally we note that the RAG value is higher for the supervised models than self-supervised for both training and test datasets.

C.4. RAG Experimental Details

In our experiments, we permute the data and iterate through mini-batches of data. For the second sampled minibatch, we can use the batch from the previous iteration to estimate the loss of the perturbed weights ϕ^{adv}. This is an unbiased sample of the loss on the remainder of the dataset. We found it sufficient to iterate through the entire dataset once for our experiments, though the computation can be made more precise with multiple epochs. Computationally, we do two forward passes through the data and one backward pass, so computing RAG is marginally more expensive than 1 epoch of training. This is significantly cheaper in terms of wall clock time relative to running the Lanzcos algorithm.

C.5. Hyperparameter details

To compare the results of the fine-tuned downstream classifier from the pre-trained ResNet-18 architecture using SimCLR, For the random feature extractor (random-ResNet-18). We do a grid search with three seeds, using SGD with an initial learning rate $\in \{0.01, 0.03\}$, 0.9 momentum, and batch size $\in \{32, 128, 512\}$.

We also ran experiments with a convolutional network autoencoder that is trained to minimize a mean squared reconstruction loss. However, using the encoder representations for fine-tuning only performed slightly better than the random features. We suspect the autoencoder performance can be improved with more recent regularization techniques such as masking and leave the comparison for future work.

D. Limitations

We are interested in investigating the impact of flatness on ability of a model to generalize, however we did not causally test this hypothesis. Although we find that models trained with self-supervised learning do have flatter landscapes as shown by adversarial gradient and measuring eigenspectrum of the hessian, it is unclear whether the relationship is causal. In addition in this work we reason about self-supervised model optima flatness but perform evaluations only on SimCLR. Our work can benefit from additional method investigations such as BYOL, VicReg, and MoCo (Grill et al., 2020; Bardes et al., 2021; He et al., 2020) and from experiments in other domains and datasets. We aim to address these in the next version of the work.