Escaping Saddle Points for Effective Generalization on Class-Imbalanced Data

Harsh Rangwani* **Sumukh K Aithal*** **Mayank Mishra** **R. Venkatesh Babu**
Video Analytics Lab, Indian Institute of Science, Bengaluru, India
{harshr@iisc.ac.in, sumukhaithal6@gmail.com,
mayankmishra@iisc.ac.in, venky@iisc.ac.in}

Abstract

Real-world datasets exhibit imbalances of varying types and degrees. Several techniques based on re-weighting and margin adjustment of loss are often used to enhance the performance of neural networks, particularly on minority classes. In this work, we analyze the class-imbalanced learning problem by examining the loss landscape of neural networks trained with re-weighting and margin based techniques. Specifically, we examine the spectral density of Hessian of class-wise loss, through which we observe that the network weights converges to a saddle point in the loss landscapes of minority classes. Following this observation, we also find that optimization methods designed to escape from saddle points can be effectively used to improve generalization on minority classes. We further theoretically and empirically demonstrate that Sharpness-Aware Minimization (SAM), a recent technique that encourages convergence to a flat minima, can be effectively used to escape saddle points for minority classes. Using SAM results in a 6.2% increase in accuracy on the minority classes over the state-of-the-art Vector Scaling Loss, leading to an overall average increase of 4% across imbalanced datasets. The code is available at https://github.com/val-iisc/Saddle-LongTail.

1 Introduction

In recent years, there has been a lot of progress in visual recognition thanks to the availability of well-curated datasets [32, 42], which are artificially balanced in terms of the frequency of samples across classes. However, modern real-world datasets are often imbalanced (*i.e.* long-tailed etc.) [31, 46, 47] and suffer from various kinds of distribution shifts. It has been shown that overparameterized models like Deep Neural Networks usually overfit classes with a high frequency of samples ignoring the minority (tail) ones [6, 47]. In such a scenario, when evaluated on metrics that focus on performance on minority data, these models perform poorly. These metrics are an essential and practical criterion for evaluating models in various domains like fairness [12], medical imaging [54] etc.

Many approaches used for improving the generalization performance of models trained using imbalanced data are based on the re-weighting of loss [14]. The weights for samples of each class are determined so that the expected loss closely approximates the testing criterion objective. In recent years, re-weighting techniques such as Deferred Re-Weighting (DRW) [8], and Vector Scaling (VS) Loss [30] have been introduced, which improve the classical re-weighting method of weighing loss proportional to the inverse of class frequency. However, even these improved re-weighting techniques lead to overfitting on the samples of tail classes. Also, it has been shown that re-weighted loss for deep networks lead to convergence to final solutions similar to the un-weighted loss, rendering it ineffective in practice [7].

*Equal Contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

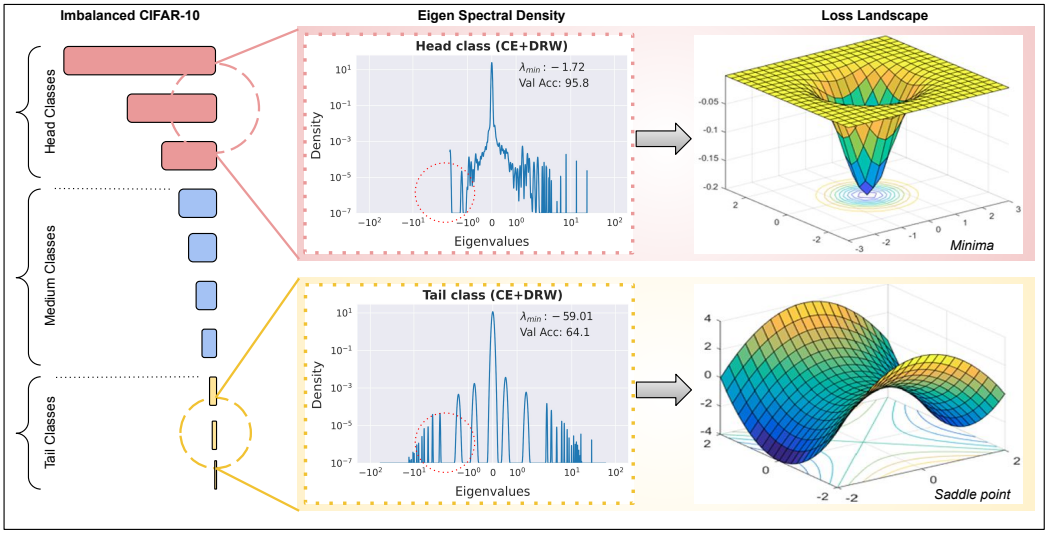

Figure 1: With class-wise Hessian analysis of loss, we observe that when deep neural networks are trained on class-imbalanced datasets, the final solution for tail classes reach a region of large negative curvature indicating convergence to saddle point (bottom), whereas the head classes converge to a minima (top). The properties of the loss landscape (saddle points or minima) can be observed by analyzing eigen spectral density (centre). [2]

This work looks at the loss landscape in weight space around final solutions attained by training with re-weighted loss. We find that the generic Hessian-based analysis of the average loss does not uncover any insights into the suboptimal generalization on tail classes (Sec. 3). As the frequency is different for each class due to imbalance, we look at the Hessian of the loss for each class. This new way of analysis finds that re-weighting cannot prevent convergence to saddle points with high negative curvature for tail classes, which eventually leads to poor generalization [16]. Whereas for head classes, the solutions converge to a minima with almost no significant presence of negative curvature, similar to networks trained on balanced data. The problem of converging to saddle points has not received much traction in recent times, as the negative eigenvalues disappear when trained on balanced datasets, indicating convergence to local minima [9, 19]. However, our analysis shows that convergence to saddle points is still a practical problem for neural networks when they are trained on imbalanced (long-tailed) data (Fig. 1).

In the community, a lot of effort has been made to develop optimization methods that are able to escape saddle points efficiently [18, 25, 26], which involve adding a component of isotropic noise to gradients. However, these methods have not been able to improve the performance of deep networks effectively, as the implicit noise of SGD in itself mitigates the issue of saddle points when trained on balanced data [15, 26]. In the case of imbalanced datasets, we find that the component of SGD along negative curvature (i.e., implicit noise) is insufficient to escape saddle points for minority classes. Thus, learning on imbalanced data can be used as a practical benchmark for optimization algorithms. We further show that Sharpness-Aware Minimization (SAM) with re-weighting can effectively enhance the gradient component along the negative curvature, allowing effective escape from saddle points.

We find that SAM can significantly improve the performance of various re-weighting and margin enhancing methods introduced for long-tailed learning. The significant improvement is also similarly seen on large-scale datasets of ImageNet-LT and iNaturalist 2018, demonstrating our observations' applicability at scale. We summarize our contributions below:

- We propose class-wise Hessian analysis of loss which reveals convergence to saddle points in the loss landscape for minority classes. We find that even loss re-weighting solutions converge to saddle point, leading to sub-optimal generalization on the minority classes.
- We theoretically show that SAM with re-weighting and high regularization significantly enhances the component of Stochastic Gradient along the direction of negative curvature enabling effective escape from saddle points.

- We find that SAM can successfully enhance the performance of even state-of-the-art techniques for learning on imbalanced datasets which have a re-weighting component (*e.g.* VS Loss and LDAM) across various datasets and degrees of imbalance.

2 Related Work & Background

In this work, we use $g(x)$ to denote the output of a model, $\nabla g(x)$ to denote the gradients with respect to parameters, x and y denote the data and labels, respectively. We review the re-weighting methods used for training on imbalanced data with distribution shifts, followed by optimization techniques related to our work.

2.1 Long-Tailed Learning

Re-sampling [6] and Re-weighting [21] are the most commonly used methods to train on class-imbalanced datasets. Oversampling the minority classes [10] and undersampling the majority classes [6] are two approaches to re-sampling. Oversampling leads to overfitting on the tail classes, and undersampling discards a large amount of data, which inevitably results in poor generalization. Kang et al. [27] proposed to decouple representation learning and classifier training to improve performance with the same. Mixup Shifted Label-Aware Smoothing model (MiSLAS) [57] aims to improve the calibration of models trained on long-tailed datasets by mixup and label-aware smoothing and thereby improve performance. RIDE [49] and TADE [56] are ensemble-based methods that achieve state-of-the-art on the long-tailed visual recognition. Samuel and Chechik [44] introduces a new loss, DRO-LT, based on distributionally robust optimization for learning balanced feature representations. We explore the problem of training class-imbalanced datasets through the lens of optimization and loss landscape. We will now describe some representative recent effective methods in detail, which we will use as baselines. Additional discussion on long-tailed learning methods is present in App. H.

LDAM [8]: LDAM introduces optimal margins for each class based on reducing the error through a generalization bound. It results in the following loss function where Δ_j is the margin for each class:

$$\mathcal{L}_{\text{LDAM}}(y; g(x)) = -\log \frac{e^{g(x)_y - \Delta_y}}{e^{g(x)_y - \Delta_y} + \sum_{j \neq y} e^{g(x)_j}} \quad \text{where} \quad \Delta_j = \frac{C}{n_j^{1/4}} \text{ for } j \in \{1, \ldots, k\} \tag{1}$$

The core idea of LDAM is to regularize the classes with low frequency (low *i.e.* n_j) more, in comparison to the head classes with high frequency.

DRW [8]: Deferred Re- Weighting refers to training the model with average loss till certain epochs (K), then introducing weight w_j proportional to $1/n_j$ to loss term specific to each class j at a later stage. This way of re-weighting has been shown to be effective for improving generalization performance when combined with various losses such as Cross Entropy (CE), LDAM etc. We will be using CE+DRW method as a representative re-weighting method for our analysis. We define CE+DRW loss below for completeness:

$$\mathcal{L}_{\text{CE}}(y; g(x)) = -w_y \log(e^{g(x)_y} / \sum_{j=1}^{k} e^{g(x)_y}) \text{ where } w_j = \frac{1}{1 + (n_j - 1)\mathbb{1}_{epoch \geq K}} \tag{2}$$

VS[30]: Vector Scaling loss is a recently proposed loss function which unifies the idea of multiplicative shift (CDT shift [53]), additive shift (i.e Logit Adjustment [38]) and loss re-weighting. The final loss has the following form:

$$\mathcal{L}_{\text{VS}}(y; g(x)) = -w_y \log(e^{\gamma_y g(x)_y + \Delta_y} / \sum_{j=1}^{k} e^{\gamma_j g(x)_j + \Delta_j}) \tag{3}$$

Here the γ_j and Δ_j are the multiplicative and additive logit hyperparameters, respectively.

[2]Figures for the minima and saddle point are from [3] and used for illustration purposes only.

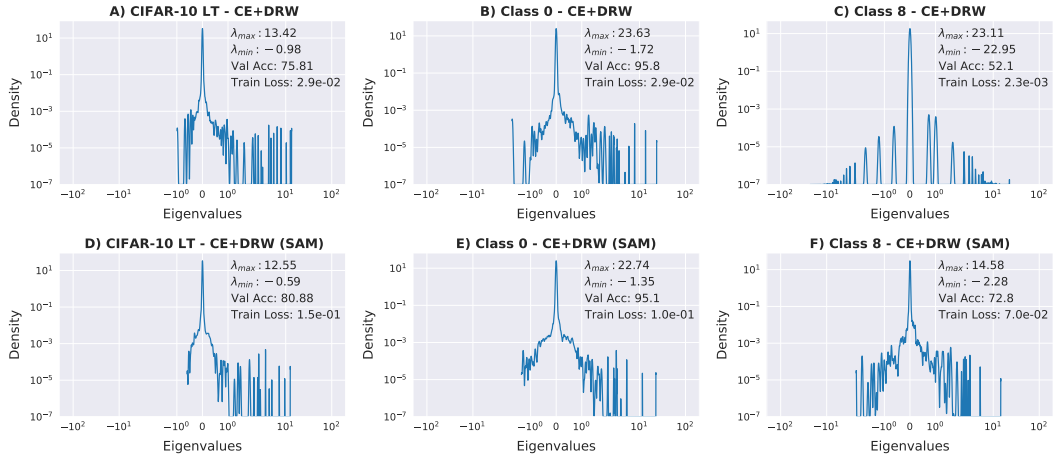

Figure 2: Eigen Spectral Density (Class-wise) on the head class (Class 0) and tail class (Class 8) with SGD and SAM. It can be observed that with the head classes, the validation accuracy with SGD (B) and SAM (E) are similar and the density of negative eigenvalues is not significant. On the tail class, SAM (F) escapes the saddle points (large density of negative eigenvalues) in SGD (C), leading to 20% increase in validation accuracy. A and D show the overall spectral density calculated across all samples in the dataset. Overall spectral density does not indicate the presence of saddles.

2.2 Loss Landscape

Saddle Points: Saddle points are regions in loss landscape that usually depict a plateau region with some negative curvature. In the non-convex setting, it has been shown that there is an existence of an exponential number of saddle points in loss landscape and convergence to these points demonstrate poor generalization [16]. There has been a lot of effort in developing methods for effectively escaping saddle points which involve the addition of noise (e.g., Perturbed Gradient Descent (PGD) [18, 25, 26]). However, these algorithms have not received much attention in the deep learning community as it has been shown that the implicit noise in SGD can escape saddles easily and converge to local minima [15]. Also, it has been empirically shown that negative eigenvalues from the Hessian spectrum disappear after a few steps of training, indicating escape from saddle points when neural networks are trained on balanced datasets [1, 9, 43]. However, contrary to this, we demonstrate that convergence to saddle points is prevalent in minority class loss landscapes and is a practical problem that can serve as a practical benchmark for the development of algorithms that escape saddle points.

Flat Minima based Optimization methods: Empirically, it has been shown that converging to a flat minima in loss landscape for a deep network leads to improved generalization in comparison to sharp minima [24, 28]. Recent works have tried to exploit this connection between the geometry of the loss landscape and generalization to achieve lower generalization error. Sharpness-Aware Minimization (SAM) [17] is one such algorithm that aims to simultaneously minimize the loss value and sharpness of the loss surface. Low-Pass Filtering SGD (LPF-SGD) [5] is another recently proposed optimization algorithm that aims to recover flat minima from the optimization landscape. LPF-SGD convolves the loss function with a Gaussian Kernel with variance proportional to the norm of the parameters of each filter in the network. In this work, we aim to explore the effectiveness of such algorithms for the task of escaping saddle points, which is a new direction for these algorithms.

3 Convergence to Saddle Points in Tail Class Loss Landscape

This section analyzes the dynamics of the loss landscape of neural networks trained on imbalanced datasets. We use the Cross Entropy (CE) loss $\hat{\mathcal{L}}_{\text{CE}}$ to denote the average cross entropy loss for each class. For fine-grained analysis, we focus on average loss on each class $\hat{\mathcal{L}}_{\text{CE}}(y)$. We visualize the loss landscape of the head and tail classes through the computation of Hessian Eigenvalue Density [19]. The Hessian of the train loss for each class $H = \nabla_w^2 \hat{\mathcal{L}}_{\text{CE}}(y)$ contains important properties about the curvature of the classwise loss landscape. The Hessian Eigenvalue Density provides all suitable information regarding the eigenvalues of H. In this work, we focus on λ_{max}(max eigenvalue) and

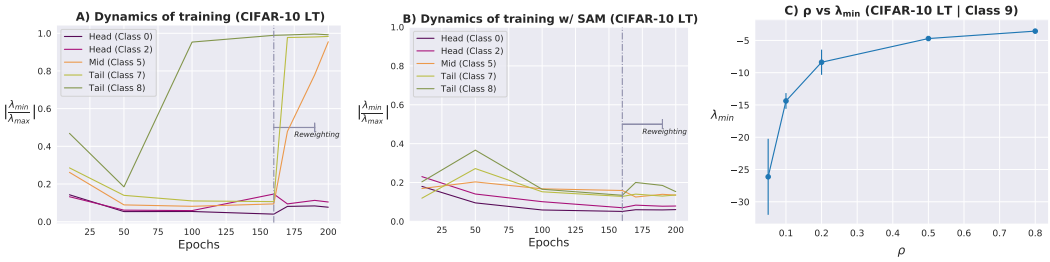

Figure 3: A) In CE+DRW, the tail class loss landscapes show significant non-convexity as indicated by the large value of $|\lambda_{min}/\lambda_{max}|$ whereas head classes (0,2) converge to convex landscapes. B) When CE+DRW is trained with SAM, it avoids convergence to non-convex regions throughout the training, as indicated by the low value of $|\lambda_{min}/\lambda_{max}|$. C) With high ρ, the $\lambda_{\min}$ increases, and the model converges to a point with low negative curvature (approx. minima).

λ_{min}(min eigenvalue), which depict the extent of positive and negative curvature present. We use the Lanczos algorithm as introduced in Ghorbani et al. [19] to compute the Hessian Eigenvalue Density (spectral density) tractably. We further calculate the validation accuracy of a particular class y and its eigen spectral density for analysis. We provide more details for these experiments in the App. D.

Does the proposed class-wise analysis of loss landscape offer any additional insights? In prior works [19, 20, 35], the Hessian of the average loss is used to characterize the nature of the converged point in the loss landscape. However, we find that when particularly trained on imbalanced datasets like CIFAR-10 LT, the eigen spectral density of the Hessian of average loss (Fig. 2A) does not differ from that of head class loss (Fig. 2B), indicating convergence to a local minima. However, explicitly analyzing the Hessian for the tail class loss (Fig. 2C) gives the correct indication of the presence of negative eigenvalues (i.e., curvature), which is in contrast to average loss. Hence, our proposed class-wise analysis of Hessian is essential for characterizing the nature of the converged solution when the training data is imbalanced.

What happens when you train a neural network with CE-DRW method on CIFAR-10 LT? Fig. 2 shows the spectral density on samples from the head class (Class 0 with 5000 samples) and tail class (Class 8 with 83 samples) at the checkpoint with the best validation accuracy. The spectral density of the head class contains few negative eigenvalues. Most of the eigenvalues are centered around zero, as also observed when training on a balanced dataset [19]. On the other hand, for the tail class, there exists a large number of both negative and positive eigenvalues, indicating convergence to a saddle point. We find that at this point, the $\hat{\mathcal{L}}_{CE}(y)$ is low along with the norm of gradient, which indicates a stationary saddle point. We also observe that the spectral density of the tail class contains many outlier eigenvalues, and λ_{max} is much larger compared to the head class indicating sharp curvature. These evidences show that *the tail class solution converges to a saddle point instead of a local minimum*. Merkulov and Oseledets [39] indicated the existence of stationary points with low error but poor generalization in the loss landscape of neural networks. Also, the existence of saddle points being associated with poor generalization has been observed for small networks [16]. However, in this work, we show that convergence to saddle points can specifically occur in the loss landscape of tail classes even for the popular ResNet [22] family of networks, which is an important and novel observation to the best of our knowledge.

Dynamics of training on Long-Tailed Datasets: We analyze the $|\lambda_{min}/\lambda_{max}|$ for the head, mid and tail classes at various epochs (10, 50, 160, 170, 190, 200) across training to understand the dynamics of optimization with CE+DRW on long-tailed data (Fig. 3A). $|\lambda_{min}/\lambda_{max}|$ is a measure of non-convexity of the loss landscape [33], where a high value of $|\lambda_{min}/\lambda_{max}|$ conveys non-convexity indicating convergence to points with significant negative curvature. The network converges to non-convex regions with negative curvature for tail classes, showing convergence to the saddle point. Also, we find that for the certain tail (Class 7, 8) and mid classes (Class 5), the network starts converging towards regions with negative curvature after applying loss re-weighting (DRW at 160th epoch). This indicates that DRW leads to convergence to a saddle point rather than preventing it.

4 Escaping Saddle Points for Improved Generalization

In this section, we analyze the Sharpness-Aware Minimization technique for escaping from saddle points in tail class loss landscape. In existing works [2, 36, 59], the effectiveness of SAM in escaping saddle points has not been explored to the best of our knowledge.

Sharpness-Aware Minimization (SAM): Sharpness-Aware Minimization is a recent technique which aims to flatten the loss landscape by first finding a sharp maximal point ϵ in the neighborhood of current weight w. It then minimizes the loss at the sharp point $(w + \epsilon)$.

$$\min_{w} \max_{||\epsilon|| \leq \rho} f(w + \epsilon) \tag{4}$$

Here, f is any objective (eg. CE or LDAM loss function) and ρ is the hyperparameter that controls the extent of neighborhood. A high value of ρ leads to convergence to much flat loss landscape. The inner optimization in above objective is first approximated using a first order solution:

$$\hat{\epsilon}(w) \approx \arg\max_{||\epsilon|| \leq \rho} f(w) + \epsilon^T \nabla f(w) \ = \rho \nabla f(w) / ||\nabla f(w)||_2 \tag{5}$$

After finding $\hat{\epsilon}(w)$, the network weights are updated using the gradient $\nabla f(w)|_{w+\hat{\epsilon}(w)}$. In recent work [2], it has been shown that the normalization of the norm of the gradient for $\hat{\epsilon}(w)$ calculation above leads to oscillation which implies non-convergence theoretically. Also, it has been empirically shown that the unnormalized version of the gradient with adjusted ρ performs better than the normalized version. Hence, we use the approximation i.e. $\hat{\epsilon}(w) = \rho \nabla f(w)$ for our theoretical results. As we will be using the stochastic version of the gradient, we use z as the stochasticity parameter and denote the gradient as $\nabla f_z(w)$. With this, we now define the gradient with respect to w that is associated with SAM:

$$\nabla f_z^{\text{SAM}}(w) = \nabla f_z(w + \hat{\epsilon}(w)) = \nabla f_z(w + \rho \nabla f_z(w)) \tag{6}$$

As we are using the same batch for obtaining the gradient to calculate the $\hat{\epsilon}(w)$ and loss, we can use the same z as the argument. We now analyze the component of the SAM gradient in the direction of negative curvature, which is required for escaping saddle points [15].

4.1 Analysis of SAM for Escaping Saddle Points

Our analysis is based on the Correlated Negative Curvature (CNC) assumption [15] that states that stochastic gradients have components along the direction of negative curvature, which helps them escape from saddle points. This assumption has been shown to be theoretically valid for the problem of learning half-spaces and also has been empirically verified for a large number of neural networks of different sizes [15]. We now formally state the assumption below:

ASSUMPTION 1 (Correlated Negative Curvature [15])**.** Let $\mathbf{v_w}$ be the minimum eigenvector corresponding to the minimum eigenvalue of the Hessian matrix $\nabla^2 f(w)$. The stochastic gradient $\nabla f_{\boldsymbol{z}}(w)$ satisfies the CNC assumption if the second moment of the projection along the direction $\mathbf{v_w}$ is uniformly bounded away from zero, i.e.

$$\exists\, \gamma \geq 0 \ s.t. \ \forall w : \mathbf{E}[< \mathbf{v_w}, \nabla f_{\boldsymbol{z}}(w) >^2] \geq \gamma \tag{7}$$

It has also been emphasized that the value of γ is shown to correlate with the magnitude of $\lambda_{\min}^2$. This shows that with a high negative eigenvalue, there is a large component of gradient along the negative curvature along $\boldsymbol{v_w}$. This allows the SGD algorithms to escape the saddle points. However, we find that in the case of class imbalanced learning (Fig. 1) even stochastic gradients may have an insufficient component in the direction of negative curvature to escape the saddle points. We now show that SAM technique, which aims to reach a flat minima, further amplifies the gradient component along negative curvature and can be effectively used to escape the saddle point. We now formally state our theorem based on the CNC assumption below:

THEOREM 2. *Let $\mathbf{v_w}$ be the minimum eigenvector corresponding to the minimum eigenvalue $\lambda_{\min}$ of the Hessian matrix $\nabla^2 f(w)$. The $\nabla f_{\boldsymbol{z}}^{SAM}(w)$ satisfies that it's second moment of projection in v_w is atleast $(1 + \rho\lambda_{min})^2$ times the original (component of $\nabla f_{\boldsymbol{z}}(w)$):*

$$\exists\, \gamma \geq 0 \ s.t. \ \forall w : \mathbf{E}[< \mathbf{v_w}, \nabla f_{\boldsymbol{z}}^{SAM}(w) >^2] \geq (1 + \rho\lambda_{min})^2 \gamma \tag{8}$$

Table 1: Results on CIFAR-10 LT and CIFAR-100 LT with β=100. SAM with re-weighting is able to avoid the regions of negative curvature, leading to major gain in performance on the mid and tail classes with CE, LDAM and VS.

	CIFAR-10 LT				CIFAR-100 LT			
	Acc	Head	Mid	Tail	Acc	Head	Mid	Tail
CE	$71.7_{\pm 0.1}$	$90.8_{\pm 3.6}$	$71.9_{\pm 0.4}$	$52.3_{\pm 3.7}$	$38.5_{\pm 0.5}$	$64.5_{\pm 0.7}$	$36.8_{\pm 1.0}$	$8.2_{\pm 1.0}$
CE + SAM	$73.1_{\pm 0.3}$	$93.3_{\pm 0.2}$	$74.1_{\pm 0.6}$	$51.7_{\pm 1.0}$	$39.6_{\pm 0.6}$	$66.5_{\pm 0.7}$	$38.1_{\pm 1.1}$	$8.0_{\pm 0.6}$
CE + DRW [8]	$75.5_{\pm 0.2}$	$91.6_{\pm 0.4}$	$74.1_{\pm 0.4}$	$61.4_{\pm 0.9}$	$41.0_{\pm 0.6}$	$61.3_{\pm 1.3}$	$41.7_{\pm 0.5}$	$14.7_{\pm 0.9}$
CE + DRW + SAM	$80.6_{\pm 0.4}$	$91.4_{\pm 0.3}$	$78.0_{\pm 0.4}$	$73.1_{\pm 0.9}$	$44.6_{\pm 0.4}$	$61.2_{\pm 0.8}$	$47.5_{\pm 0.6}$	$20.7_{\pm 0.6}$
LDAM + DRW [8]	$77.5_{\pm 0.5}$	$91.1_{\pm 0.8}$	$75.7_{\pm 0.7}$	$66.4_{\pm 0.2}$	$42.7_{\pm 0.3}$	$61.8_{\pm 0.6}$	$42.2_{\pm 1.5}$	$19.4_{\pm 0.9}$
LDAM + DRW + SAM	$81.9_{\pm 0.4}$	$91.0_{\pm 0.2}$	$79.2_{\pm 0.5}$	$76.4_{\pm 1.1}$	$45.4_{\pm 0.1}$	$64.4_{\pm 0.3}$	$46.2_{\pm 0.2}$	$20.8_{\pm 0.3}$
VS [30]	$78.6_{\pm 0.3}$	$90.6_{\pm 0.4}$	$75.8_{\pm 0.5}$	$70.3_{\pm 0.5}$	$41.7_{\pm 0.5}$	$54.4_{\pm 0.2}$	$41.1_{\pm 0.6}$	$26.8_{\pm 1.0}$
VS + SAM	$82.4_{\pm 0.4}$	$90.7_{\pm 0.0}$	$79.6_{\pm 0.5}$	$78.0_{\pm 01.2}$	$46.6_{\pm 0.4}$	$56.4_{\pm 0.4}$	$48.8_{\pm 0.6}$	$31.7_{\pm 0.1}$

REMARK. The above theorem adds the factor $(1+\rho\lambda_{\min})^2$ to increase the component in direction of negative curvature (γ) when $\lambda_{\min} \leq \frac{-2}{\rho}$. Due to this increase, the model will be able to escape from directions with high negative curvature, leading to an increased $\lambda_{\min}$. Also, as the factor $\frac{-2}{\rho}$ is inversely proportional to ρ, the high value of ρ aids in effectively increasing the minimum negative eigenvalue. To empirically verify this, we evaluate the Hessian spectrum for the CIFAR-10 LT dataset using CE-DRW method for different values of ρ (Fig. 3C). We find that, as expected from the theorem, in practice, the high values of ρ lead to less negative values of $\lambda_{\min}$. This indicates escaping the saddle points effectively, hence avoiding convergence to regions having negative curvature in loss landscape. The proof of the above theorem and additional details is provided in Appendix B.

We also want to convey that theoretically, techniques like Perturbed Gradient Descent (PGD), and LPF-SGD (Low-Pass Filter SGD), which add Gaussian noise into gradient to escape saddle points can also be used for mitigating negative curvature. Also it has been found that SGD [15] can also escape the saddle points and converges to solutions with a flat loss landscape. Also, theoretically according to Theorem 2 in Daneshmand et al. [15] the SGD algorithm convergence to a second-order stationary point depends on the γ as $\mathcal{O}(\gamma^{-4})$ under some assumptions on f. As we find that as SAM with high ρ enhances the component of SGD in direction of negative curvature (γ) by $(1+\rho\lambda_{\min})^2$, it is reasonable to expect that SAM is able to escape saddle points effectively and converge to solutions with significant less negative curvature implying better generalization. We provide empirical evidence for this in Fig. 3B and Sec. 5.2.

What happens when you train a neural network with SAM + DRW? With SAM (high ρ), the large negative eigenvalues present in the loss landscape of the tail class get suppressed (Fig. 2F). In the spectral density for the tail class, it can be seen that λ_{min} is much closer to zero for SAM compared to its counterpart with SGD. This aligns with the hypothesis that SAM escapes regions of negative curvature, leading to improved accuracy on the tail classes. However, the spectral density of the head class does not change significantly compared to that of Empirical Risk Minimization (ERM), although the λ_{max} is much lower for SAM, indicating a flatter minima for the head class.

We also analyze the $|\lambda_{min}/\lambda_{max}|$ across multiple steps of training with SAM (Fig. 3B), where $|\lambda_{min}/\lambda_{max}|$ is a measure of non-convexity of the loss surface. We observe that SAM does not allow the tail classes to reach a region of high non-convexity. The values of $|\lambda_{min}/\lambda_{max}|$ is much lower for SAM compared to SGD (Fig. 3A) throughout training, indicating minimal negative eigenvalues (*i.e.* more convexity) in the loss landscape, especially for the tail and medium classes. This clearly shows that SAM avoids regions of substantial negative curvature in the search of flat minima. Further, we note that once the re-weighting begins, SAM is able to avoid convergence to a saddle point (non-convexity decreases), which is contrary to what we observe with CE+DRW (with SGD). Theorem 3 states that SAM consists of larger component in the direction of negative curvature which allows to reach a solution with minimal negative curvature. Empirically, Fig. 3B also supports the Theorem 3 as we observe that SAM reaches a minima (high convexity) for all the classes.

Table 2: Results on CIFAR-10 and CIFAR-100 with Step Imbalance (β = 100). SAM generalizes well across datasets with varied type of imbalance, resulting in substantial gain in tail accuracy in all settings.

	CIFAR-10			CIFAR-100		
	Acc	Head	Tail	Acc	Head	Tail
CE	65.1	88.6	41.7	38.6	76.3	00.9
CE + SAM	66.1	92.9	39.4	39.3	78.6	00.0
CE + DRW [8]	72.2	93.1	51.2	45.8	73.9	17.8
CE + DRW + SAM	79.3	92.7	65.8	48.3	73.1	23.4
LDAM + DRW [8]	77.6	89.2	66.0	45.3	70.3	20.4
LDAM + DRW + SAM	81.0	90.5	71.5	49.2	74.0	24.4
VS [30]	77.0	91.7	62.3	46.5	69.0	24.1
VS + SAM	82.0	91.7	72.3	48.3	70.4	26.2

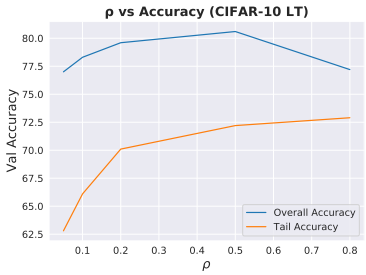

Figure 4: Impact of ρ on Overall Accuracy and Tail Accuracy (CIFAR-10 LT).

5 Experiments

5.1 Class-Imbalanced Learning

Datasets: We report our results on four long-tailed datasets: CIFAR-10 LT [8], CIFAR-100 LT [8], ImageNet-LT [37], and iNaturalist 2018 [48]. **a) CIFAR-10 LT and CIFAR-100 LT**: The original CIFAR-10 and CIFAR-100 datasets consist of 50,000 training images and 10,000 validation images, spread across 10 and 100 classes, respectively. We use two imbalance versions, i.e., long-tail imbalance and step imbalance, as followed in Cao et al. [8]. The imbalance factor, $\beta = \frac{N_{\max}}{N_{\min}}$, denotes the ratio between the number of samples in the most frequent ($N_{\max}$) and least frequent class ($N_{\min}$). For both the imbalanced versions, we analyze the results with β = 100. **b) ImageNet-LT and iNaturalist 2018**: We use the ImageNet-LT version as proposed by [37], which is an class-imbalanced version of the large-scale ImageNet dataset [42]. It consists of 115.8K images from 1000 classes, with 1280 images in the most frequent class and 5 images in the least. iNaturalist 2018 [48] is a real-world long-tailed dataset that contains 437.5K images from 8,142 categories. In the case of long-tail imbalance, we segregate the classes of all the datasets into *Head* (Many), *Mid* (Medium), and *Tail* (Few) subcategories, as defined in [57]. For step imbalance experiments on CIFAR datasets, we split the classes into *Head* (Frequent) and *Tail* (Minority), as done in [8].

Experimental Details: We follow the hyperparameters and setup as in Cao et al. [8] for CIFAR-10 LT and CIFAR-100 LT datasets. We train a ResNet-32 architecture as the backbone and SGD with a momentum of 0.9 as the base optimizer for 200 epochs. A multi-step learning rate schedule is used, which drops the learning rate by 0.01 and 0.0001 at the 160th and 180th epoch, respectively. For training with SAM, we set a constant ρ value of either 0.5 or 0.8 for most methods. For ImageNet-LT and iNaturalist 2018 datasets, we use the ResNet-50 backbone similar to [57]. An initial learning rate of 0.1 and 0.2 is set for iNaturalist 2018 and ImageNet-LT, respectively, followed by a cosine learning rate schedule. We initialize the ρ value with 0.05 and utilize a step schedule to increase the ρ value during the course of training for SAM experiments. We run every experiment on long-tailed CIFAR datasets with three seeds and report the mean and standard deviation. Additional implementation details are provided in the App. C. Algorithm for DRW+SAM is defined in App. G.

Baselines: **a) Cross-Entropy (CE)**: CE minimizes the average loss across all samples, and thus, the performance of tail classes is much lower than that of head classes. **b) CE + Deferred Re-Weighting (DRW) [8]**: The re-weighting of CE loss inversely by class frequency is done in the later stage of training. **c) LDAM + DRW [8]**: Label-Distribution-Aware Margin (LDAM) proposes a margin-based loss that encourages larger margins for less-frequent classes. **d) Vector Scaling (VS) Loss [30]**: VS loss incorporates both additive and multiplicative logit adjustments to modify inter-class margins.

Results: Table 1 summarizes our results on CIFAR-10 LT and CIFAR-100 LT with β of 100. It can be observed that SAM with re-weighting significantly improves the accuracy on mid and tail classes while preserving the accuracy on head classes. DRW + SAM improves upon the overall performance of CE+DRW by 5.1% on CIFAR-10 LT and 3.6% on CIFAR-100 LT datasets, with the tail class accuracy increasing by 11.7% and 7.7% respectively. As explained in Sec. 3, the reason for the

Table 3: Results on iNaturalist 2018 and ImageNet-LT datasets with LDAM+DRW and comparison with other methods. The numbers for methods marked with † are taken from [57].

Method		iNaturalist 2018				ImageNet-LT			
	Two stage	Acc	Head	Mid	Tail	Acc	Head	Mid	Tail
CE	×	60.3	72.8	62.7	54.8	42.7	62.5	36.6	12.5
cRT [27] †	✓	68.2	73.2	68.8	66.1	50.3	62.5	47.4	29.5
LWS [27] †	✓	69.5	71.0	69.8	68.8	51.2	61.8	48.6	33.5
MiSLAS [57]	✓	**71.6**	73.2	**72.4**	70.4	52.7	61.7	51.3	**35.8**
DisAlign [55]	✓	69.5	61.6	70.8	69.9	52.9	61.3	**52.2**	31.4
DRO-LT [44]	×	69.7	**73.9**	70.6	68.9	**53.5**	**64.0**	49.8	33.1
CE + DRW	×	63.0	59.8	64.4	62.3	44.9	57.9	42.2	21.6
CE + DRW + SAM	×	65.3	60.5	66.2	65.5	47.1	56.6	45.8	28.1
LDAM + DRW	×	67.5	63.0	68.3	67.8	49.9	61.1	48.2	28.3
LDAM + DRW + SAM	×	70.1	64.1	70.5	**71.2**	53.1	62.0	52.1	34.8

increase in performance can be attributed to the suppression of negative eigenvalues in the tail class spectral density.

These results empirically show that escaping saddle points with SAM leads to a notable increase in overall accuracy primarily due to the major gain in the accuracy on the tail classes. The addition of SAM to recently proposed long-tail learning methods like LDAM and VS loss leads to a significant increase in performance, which indicates that the role of SAM is orthogonal to the margin-based methods. On the other hand, SAM without re-weighting (CE+SAM) improves accuracy on the head and mid classes rather than the tail class. This can be attributed to the fact that standard ERM minimizes the average loss across all the samples without re-weighting and the weightage of tail class samples in the overall loss is minimal. Thus, the weightage of SAM in the tail classes is limited. We also report the results with various imbalance factor (β) in App. F.

We also show results with step imbalance (β = 100) on CIFAR-10 and CIFAR-100 datasets (Table 2). With step imbalance on CIFAR-10, the first five classes have 5000 samples each, and the remaining classes have 50 samples each. The addition of SAM improves the overall performance of CE+DRW on CIFAR-10 by 7.1%, with the tail class accuracy increasing by 14.6%. We observe that on most tail classes, the density of negative eigenvalues in the spectral density is much lower with SAM. This indicates that despite multiple classes with few samples, SAM with DRW can avoid the saddle points. SAM systematically improves performance with LDAM and VS loss leading to state-of-the-art performance on both CIFAR-10 and CIFAR-100 in the step imbalance setting.

Do these observations scale to large-scale datasets? We report the results on ImageNet-LT dataset in Table 3. We also compare with recent long-tail learning methods: cRT [27], MisLAS [57], DisAlign [55] and DRO-LT [44]. The observations on CIFAR-10 LT and CIFAR-100 LT hold good even on ImageNet-LT. For example, the accuracy on tail classes increases by 6.5% with the introduction of SAM on CE + DRW, which is similar to the gain observed in CIFAR-100 LT with CE + DRW. We observe that LDAM+DRW+SAM surpasses the performance of two-stage training methods including MisLAS, cRT, LWS, and DisAlign. Compared to these two-stage methods, our method is a single stage method and outperforms these two-stage methods. These observations point out that the problem of saddle points also exists in large datasets and convey that SAM is easily generalizable to large-scale imbalanced datasets without making any significant changes. On iNaturalist 2018 [48] too, the accuracy on tail classes gets boosted by more than 3% with SAM (Table 3).

Comparison with SOTA: VS loss [30] is a recently proposed margin-based method that achieves state-of-the-art performance on class-imbalanced datasets with single-stage training without strong augmentations [57], ensembles [56] or self-supervision [51]. SAM significantly improves upon the performance of VS on both CIFAR-10 LT and CIFAR-100 LT. For the practitioners, we suggest *using high ρ SAM with re-weighting or margin based methods* for effective learning on long-tailed data. We also integrate SAM with more recent IB-Loss [40] and Parametric Contrastive Learning (PaCo) [13] methods and report the results in App. E. We find that SAM is also effectively able to improve performance of these recent methods.

Table 4: Results on CIFAR-10 LT and CIFAR-100 LT with various methods that escape saddle points.

	CIFAR-10 LT				CIFAR-100 LT			
	Acc	Head	Mid	Tail	Acc	Head	Mid	Tail
CE + DRW	75.5	91.6	74.1	61.4	41.0	61.3	41.7	14.7
CE + DRW + PGD [25]	77.2	92.0	75.2	65.0	42.2	63.0	41.6	17.0
CE + DRW + LPF-SGD [5]	78.5	90.8	77.7	67.2	42.9	64.0	43.7	15.8
CE + DRW + SAM	80.6	91.4	78.0	73.1	44.6	61.2	47.5	20.7

5.2 Ablation Studies

A note on ρ value: We observe that as we increase the smoothness parameter (ρ) in SAM, the accuracy on the tail classes increases significantly (Fig. 4). The accuracy on tail classes increases from 63% for $\rho = 0.05$ to 73% for $\rho = 0.8$ on CIFAR-10 LT with CE+DRW. This can be ascribed to the correlation between λ_{min} and ρ as discussed in Sec. 4.1. As the ρ increases, the negative curvature in the tail classes disappears because SAM aims to find a flat minima with a large neighborhood with a low loss value. A very large ρ (0.8) leads to a drop in the head accuracy because it restricts the solution space of the head class, resulting in a drop in the overall accuracy. This also emphasizes that a high ρ is necessary for escaping saddle points and achieving the best results.

Other methods to escape saddle points: In Table 4, we show that other methods developed to escape saddle points, such as PGD, can be used for improving generalization on tail classes. LPF-SGD, an algorithm promoting convergence to flat landscape, inherently adds Gaussian noise to the network parameters and could be considered similar to PGD. We can see that the addition of PGD and LPF-SGD to CE+DRW leads to a substantial gain in the performance of tail classes on CIFAR-10 LT and CIFAR-100 LT. It can also be observed that CE+DRW+SAM outperforms both PGD and LPF-SGD by 2% on average. This further highlights that various methods in literature developed to escape saddle points efficiently can be directly used to improve the performance of minority classes when training on class-imbalanced datasets.

6 Conclusion

In this work, we show that training on imbalanced datasets can lead to convergence to points with sufficiently large negative curvature in the loss landscape for minority classes. We find that this is quite common when neural networks are trained with loss functions that are re-weighted or modified to enhance the focus on minority classes. Due to the occurrence of saddle points, we observe that the network suffers from poor generalization on minority classes. We propose to use Sharpness-Aware Minimization (SAM) with a high regularization factor ρ as an effective method to escape regions of negative curvature and enhance the generalization performance. We theoretically and empirically demonstrate that SAM with high ρ is able to escape saddle points faster than SGD and converge to better solutions, which is a novel observation to the best of our knowledge. We show that combining SAM with state-of-the-art techniques for learning with imbalanced data leads to significant gains in performance on minority classes. We hope that our work leads to further research in studying the effect of negative curvature in generalization as we show they are a practical issue for class-imbalanced learning using deep neural networks.

Acknowledgements: This work was supported in part by SERB-STAR Project (Project:STR/2020/000128), Govt. of India. Harsh Rangwani is supported by Prime Minister's Research Fellowship (PMRF). We are thankful for their support.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Added in appendix.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] To the best of our knowledge, we do not suspect any potential negative societal impacts of our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 4.
   (b) Did you include complete proofs of all theoretical results? [Yes] Proof is added in the appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Additional information added in Appendix
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5. Additional information added in Appendix.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See section 5. We report error bars for CIFAR 10-LT and CIFAR 100-LT dataset. For the large scale ImageNet-LT and iNaturalist 2018, we do not report error bars due to computational constraints.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Added in the Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] Added in appendix
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We use publicly available assets for our work.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Limitations of Our Work

We would like to highlight that our theoretical results are based on Daneshmand et al. [15] which verified CNC condition for small scale neural networks, verifying the CNC condition for large networks and exactly characterizing the saddle point solutions obtained by SAM for minority classes, are good directions for future work.

Also empirically, we propose to use Sharpness-Aware Minimization with high ρ for tail classes to escape from saddle points. Although the general guideline is to use a higher ρ value like 0.5 or 0.8 to achieve the best result, we do find that ρ as a hyperparameter still requires tuning to obtain the best results. We believe making SAM hyper-parameter free is an interesting direction to pursue in the future.

B Proof of Theorem

In this section, we re-state Theorem 3 and provide it's proof. The theorem analyzes the variance of stochastic gradient for SAM along the direction of negative curvature and shows that SAM amplifies the variance by a factor, which signals that it has a stronger component in direction of negative curvature under certain conditions. Hence, SAM can be used for effectively escaping saddle points in the loss landscape. This is based on Correlated Negative Curvature (CNC) Assumption for stochastic gradients (Assumption 1). The $\mathbf{v_w}, \nabla f(w) \in \mathbb{R}^{p\times 1}$ whereas the Hessian denoted by $H(f(w))$ (also$\nabla^2 f(w)) \in \mathbb{R}^{p\times p}$ where p is the number of parameters in the model.

THEOREM 3. *Let $\mathbf{v_w}$ be the minimum eigenvector corresponding to the minimum eigenvalue $\lambda_{\min}$ of the Hessian matrix $\nabla^2 f(w)$. The $\nabla f_{\boldsymbol{z}}^{SAM}(w)$ satisfies that it's second moment of projection in v_w is atleast $(1+\rho\lambda_{min})^2$ times the original (component of $\nabla f_{\boldsymbol{z}}(w)$):*

$$\exists\, \gamma \geq 0 \; s.t. \; \forall w : \mathbf{E}[< \mathbf{v_w}, \nabla f_{\boldsymbol{z}}^{SAM}(w) >^2] \geq (1+\rho\lambda_{min})^2\gamma \tag{9}$$

Proof. Using the first-order approximation of a vector valued function through Taylor series:

$$f(w+\epsilon) = f(w) + J(\nabla f(w))\epsilon \tag{10}$$

here J is the jacobian operator. After considering ρ to be small we have the following approximation for the SAM gradient:

$$\nabla f^{\text{SAM}}(\boldsymbol{w}) = \nabla f(w + \rho\nabla f(w)) \tag{11}$$

$$= \nabla f(w) + \rho H(f(w))\nabla f(w) \tag{12}$$

Here, we have used the following property that $J(\nabla f(w))$ is the Hessian matrix $H(f(w))$ (also written as $\nabla^2 f(w)$). Also, as we now want to work with stochastic gradients, we replace gradient $\nabla f(w)$ with it's stochastic version $\nabla f_z(w)$ and introduce an expectation expression. Now, we analyze the second-moment of the SAM gradient along the direction of most negative curvature $\boldsymbol{v_w}$:

$$\begin{aligned}
\boldsymbol{E}[< \mathbf{v_w}, \nabla f_{\boldsymbol{z}}^{\text{SAM}}(w) >^2] &= \boldsymbol{E}[< \mathbf{v_w}, \nabla f_z(w) + \rho H(f(w))\nabla f_z(w)) >^2] \\
&= \boldsymbol{E}[(< \mathbf{v_w}, \nabla f_z(w) > + \rho < \mathbf{v_w}, H(f(w))\nabla f_z(w) >)^2] \\
&= \boldsymbol{E}[(< \mathbf{v_w}, \nabla f_z(w) > + \rho \mathbf{v_w}^T H(f(w))\nabla f_z(w))^2]
\end{aligned}$$

Here, we use the matrix notation for dot product $< x, y >= x^T y$. Using the property of the eigen vector: $\mathbf{v_w^T} H(f(w)) = \lambda_{min}\mathbf{v_w^T}$, we substitute the value below:

$$\begin{aligned}
\boldsymbol{E}[< \mathbf{v_w}, \nabla f_{\boldsymbol{z}}^{\text{SAM}}(w) >^2] &= \boldsymbol{E}[(< \mathbf{v_w}, \nabla f_z(w) > + \rho\lambda_{min}\mathbf{v_w^T}\nabla f_z(w) >)^2] \\
&= \boldsymbol{E}[(< \mathbf{v_w}, \nabla f_z(w) > + \rho\lambda_{min} < \mathbf{v_w}, \nabla f_z(w) >)^2] \\
&= \boldsymbol{E}[((1+\rho\lambda_{min}) < \mathbf{v_w}, \nabla f_z(w) >)^2] \\
&= (1+\rho\lambda_{min})^2 \boldsymbol{E}[< \mathbf{v_w}, \nabla f_z(w) >^2] \\
&\geq (1+\rho\lambda_{min})^2\gamma
\end{aligned}$$

The last step follows from the CNC Assumption 1. This completes the proof. □

Table 5: ρ value for used for reporting the results with SAM.

	CIFAR-10		CIFAR-100	
	LT (β = 100)	Step (β = 100)	LT (β = 100)	Step (β = 100)
CE + SAM	0.1	0.1	0.2	0.5
CE + DRW + SAM	0.5	0.2	0.8	0.2
LDAM + DRW + SAM	0.8	0.1	0.8	0.5
VS + SAM	0.5	0.2	0.8	0.2

C Experimental Details

Imbalanced CIFAR-10 and CIFAR-100: For the long-tailed imbalance (CIFAR-10 LT and CIFAR-100 LT), the sample size across classes decays exponentially with $\beta = 100$. CIFAR-10 LT holds 5000 samples in the most frequent class and 50 in the least, whereas CIFAR-100 LT decays from 500 samples in the most frequent class to 5 in the least. The classes are divided into three subcategories: *Head* (Many), *Mid* (Medium), and *Tail* (Few). For CIFAR-10 LT, the first 3 classes (> 1500 images each) fall into the head classes, following 4 classes (> 250 images each) into the mid classes, and the final 3 classes (< 250 images each) into the tail classes. Whereas for CIFAR-100 LT, head classes consist of the initial 36 classes, mid classes contain the following 35 classes, and the tail classes consist of the remaining 29 classes.

In the step imbalance setting, both CIFAR-10 and CIFAR-100 are split into two classes, i.e., *Head* (Frequent) and *Tail* (Minority), with $\beta = 100$. The first 5 (Head) classes of CIFAR-10 contain 5000 samples each, along with 50 samples each in the remaining 5 (Tail) classes. On the other hand, the top first 50 (Head) classes of CIFAR-100 contain 500 samples each, and the remaining 50 (Tail) classes consist of 5 samples each.

All the experiments on imbalanced CIFAR-10 and CIFAR-100 are run with ResNet-32 backbone and SGD with momentum 0.9 as the base optimizer. All the methods train on imbalanced CIFAR-10 and CIFAR-100 with a batch size of 128 for 200 epochs, except for VS Loss, which runs for 300 epochs. We follow the learning rate schedule mentioned in Cao et al. [8]. In the initial 5 epochs, we linearly increase the learning rate to reach 0.1. Following that, a multi-step learning rate schedule decays the learning rate by scaling it with 0.001 and 0.0001 at 160th and 180th epoch, respectively. For LDAM runs on imbalanced CIFAR, the value of C is tuned so that Δ_j is normalised to set maximum margin of 0.5 (refer to Equation. 1 in main text). In the case of VS Loss, we use γ as 0.05 and τ as 0.75 for imbalanced CIFAR-10 and CIFAR-100 datasets (refer to Equation. 3 in main text).

ImageNet-LT and iNaturalist 2018: The classes in ImageNet-LT and iNaturalist 2018 datasets are also divided into three subcategories, i.e., *Head* (Many), *Mid* (Medium), and *Tail* (Few). For ImageNet-LT, the head classes consist of the first 390 classes, mid classes contain the subsequent 445 classes, and the tail classes hold the remaining 165 classes. Whereas for iNaturalist 2018, first 842 classes fall into the head classes, subsequent 3701 classes into the mid classes, and the remaining 3599 into the tail classes.

For ImageNet-LT and iNaturalist 2018, all the models are trained for 90 epochs with a batch size of 256. We use ResNet-50 architecture as the backbone and SGD with momentum 0.9 as the base optimizer. A cosine learning rate schedule is deployed with an initial learning rate of 0.1 and 0.2 for iNaturalist 2018 and ImageNet-LT, respectively. For LDAM runs on ImageNet-LT and iNaturalist 2018, the value of C is tuned so that Δ_j is normalised to set maximum margin of 0.3 (refer to Equation. 1 in main text).

Optimum ρ value: Table 5 compiles the ρ value used by SAM across various methods on imbalanced CIFAR-10 and CIFAR-100 datasets. The ρ value in these runs is kept constant throughout the duration of training. We adopt a common step ρ schedule for the SAM runs on both ImageNet-LT and iNaturalist 2018. We initialise the ρ with 0.05 for the initial 5 epochs and change it to 0.1 till the 60th epoch. Following that, we increase the ρ value to 0.5 for the final 30 epochs.

Table 6: Results on ImageNet-LT (ResNet-50) with LDAM+DRW and comparison with other methods. The numbers for methods marked with † are taken from [57].

	Two stage	Acc	Head	Mid	Tail
CE	×	42.7	62.5	36.6	12.5
cRT [27] †	✓	50.3	62.5	47.4	29.5
LWS [27] †	✓	51.2	61.8	48.6	33.5
MisLAS [57]	✓	52.7	61.7	51.3	**35.8**
DisAlign [55]	✓	52.9	61.3	**52.2**	31.4
DRO-LT* [44]	×	**53.5**	**64.0**	49.8	33.1
LDAM + DRW	×	49.9	61.1	48.2	28.3
LDAM + DRW + SAM	×	53.1	62.0	52.1	34.8

How to select ρ ? ρ is an hyperparameter in the SAM algorithm and it is important to choose the right value of ρ for best performance on long-tailed learning. We observe that default value of ρ (0.05) as suggested in Foret et al. [17] does not lead to significant gain in accuracy (Refer Fig. 4 in main paper), as it is not able to escape the region of negative curvature. On long-tail CIFAR-10 and CIFAR-100 setting with re-weighting (DRW), a large value of ρ (0.5 or 0.8) seems to work best instead, as in this work our objective to escape saddle points instead of improving generalization. This can be intuitively understood as large regularization (ρ) is required for highly imbalanced datasets to escape saddle points as suggested by Theorem 3. In Table 5, we have reported the ρ value used in every experiment. For the large scale datasets like ImageNet-LT and iNaturalist 18, we found that progressively increasing the ρ value gives the best results. This is based on the idea that, as the training progresses, more flatter regions can be recovered from the loss landscape [5]. In our experiments on ImageNet-LT, we use a large ρ of 0.5 in the last 30 epochs of training and we observe that the tail accuracy significantly increases at this stage of training. For using the proposed method on a new imbalanced dataset, we suggest starting with ρ = 0.05 and increasing ρ till the overall accuracy starts to decrease.

LPF-SGD and PGD: We use the official implementation of LPF-SGD [5] [3] to report the results on CIFAR-10 LT and CIFAR-100 LT. For LPF-SGD, we use Monte Carlo iterations (M) = 8 and a constant filter radius (γ) of 0.001 (as defined in Algorithm 4.1 in Bisla et al. [5]). We implement the stochastic PGD method [25, 26] on our own since there is no official PyTorch implementation available. We sample the perturbation (noise) from a Gaussian distribution with zero mean and (σ) standard deviation. We use a σ of 0.0001 for CIFAR-10 and CIFAR-100 LT experiments.

Hessian Experiments: For calculating the Eigen Spectral Density, we use the PyHessian library [52]. PyHessian uses Lanczos algorithm for fast and efficient computation of the complete Hessian eigenvalue density. The Hessian is calculated on the average loss of the training samples as done in [19, 52]. λ_{min} and λ_{max} are extracted from the complete Hessian eigenvalue density. It has been shown that the estimated spectral density calculated with the Lanczos algorithm can be used as an approximate to the exact spectral density [19]. Several works [17, 19, 20, 52] have used the same method to calculate spectral density and analyze the loss landscape of neural networks.

All of our implementations are based on PyTorch [41]. For experiments pertaining to imbalanced CIFAR, we use NVIDIA GeForce RTX 2080 Ti, whereas for the large scale ImageNet-LT and iNaturalist 2018, we use NVIDIA A100 GPUs. We log all our experiments with Wandb [4].

D Additional Eigen Spectral Density Plots

We find that the spectral density of a class is representative of the other classes in same category (Head, Mid or Tail), hence for brevity we only display the eigen spectrum of one class per category for analysis.

CE: The spectral density on the standard CE loss (without re-weighting) can be seen in Fig. 5. We notice that the density and magnitude of negative eigenvalues is much larger for the tail class (Class

[3] https://github.com/devansh20la/LPF-SGD

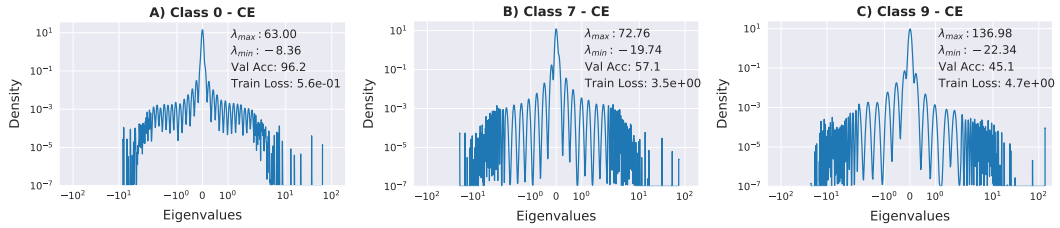

Figure 5: Eigen Spectral Density of Head (Class 0) and Tail (Class 7 and Class 9) with standard CE (without re-weighting). Since CE minimizes the average loss, it can be seen that the loss on the tail class samples (B and C) is quite high. On the head class (A), the loss is low and λ_{min} is close to 0.

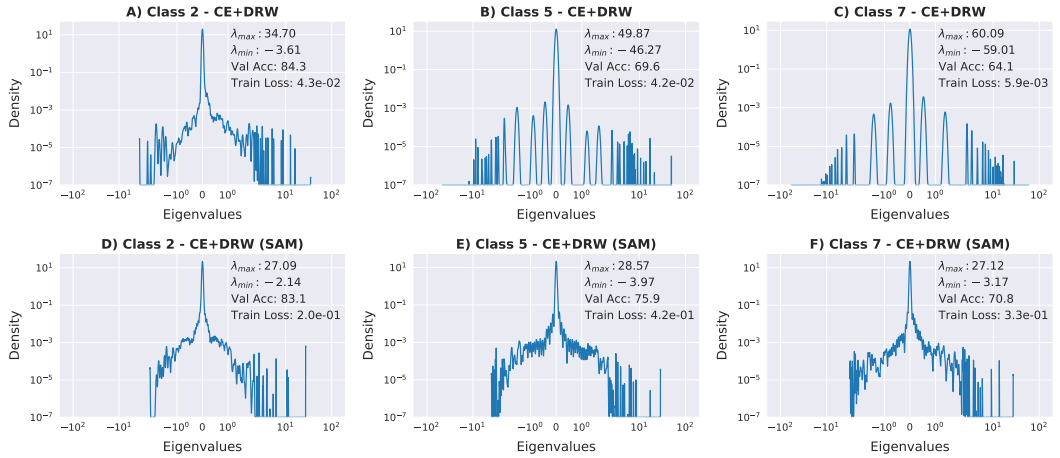

Figure 6: Eigen Spectral Density of the Head (Class 2), Mid (Class 5) and Tail classes (Class 7) with CE+DRW and CE+DRW+SAM.

7 and Class 9 in Fig. 5B and 5C) compared to the head classes (Fig. 5A). On the other hand, the spectral density of the head class (Class 0) is very different from that of the tail class, with λ_{min} of the head class very close to 0 indicating convergence to minima.

It must be noted that without re-weighting, the loss on the tail class samples is high because CE minimizes the average loss. Hence, the solution may not converge for tail class loss. However, in CE+DRW after re-weighting, we observe that the loss on tail class samples is very low, which indicates convergence to a stationary point. Thus, in CE+DRW, we can evidently conclude that the presence of large negative curvature indicates convergence to a saddle point. In summary, we find that just using CE converges to a point with significant negative curvature in tail class loss landscape. Further, though DRW is able to decrease the loss on tail classes, it still does converge to a point with significant negative curvature. This indicates that it converges to a saddle point instead of a minima. Hence, both CE and CE+DRW do not converge to local minima in tail class loss landscape.

CE+DRW: We show additional class wise Eigen Spectral Density plots with CE+DRW and CE+DRW with SAM in Fig. 6. We analyze the spectral density plots on Head (Class 2), Mid (Class 5) and Tail (Class 7). It can be seen that the magnitude of λ_{max} and λ_{min} is much lower with SAM in all the classes (Fig. 6 D, E, F). This indicates that SAM reaches a flatter local minima with no significant presence of negative eigenvalues, escaping saddle points.

LDAM: We also show Spectral density plots of Class 0 (Fig. 7 A, C) and Class 9 (Fig. 7 B, D) with LDAM+DRW method (SGD and SAM) in Fig. 7. The existence of negative eigenvalues in the tail class spectral density (Fig. 7B) indicates that even for LDAM loss (a regularized margin based loss), the solutions do converge to a saddle point. This also indicates that observations with CE+DRW hold good for long-tailed learning methods like LDAM which use margins instead of re-weighting directly. Hence, this gives evidence of the reason why SAM can be combined easily with LDAM, VS Loss etc. to effectively improve performance.

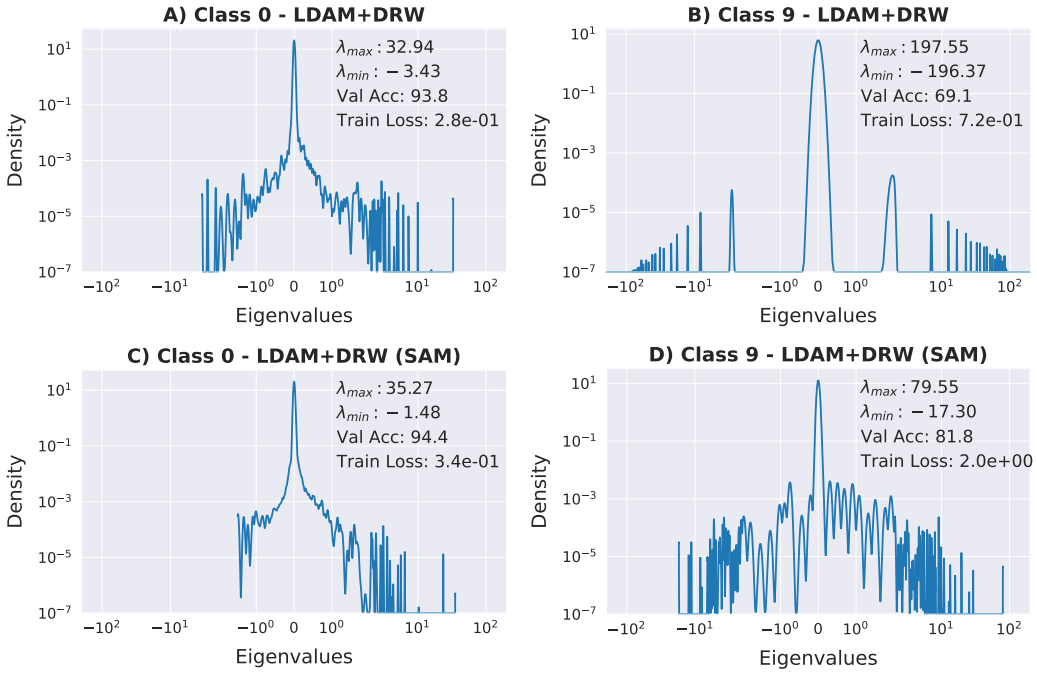

Figure 7: Eigen Spectral Density of the Head (Class 0) and Tail (Class 9) class trained with LDAM. Even with LDAM, we observe existence of negative eigenvalues in the loss landscape for the tail class, which reduce in magnitude when LDAM is used with SAM.

The spectral density of the tail class of LDAM with SAM (Fig. 7D) contains fewer negative eigenvalues compared to SGD (Fig. 7B). This indicates convergence to local minima and clearly explains why SAM improves the performance of LDAM by 12.7%.

E Additional Results

For further establishing the generality of our method, we choose two recent orthogonal method Influence-Balanced Loss [40] (IB-Loss) and Parametric Contrastive Learning (PaCo) [13] and apply proposed high ρ SAM over them. We use the open-source implementations of IB-Loss [4] and PaCo [5] to reproduce the results and add our proposed method (high ρ SAM) to that setup to obtain the results reported in the table below. We show results on CIFAR-100 LT with an imbalance factor (β) of 100 and 200. We observe that SAM with high ρ significantly improves overall performance along with the performance on tail classes with both IB-Loss and PaCo method (Table 9). Despite PaCo baseline achieving close to state-of-the-art performance, the addition of high ρ SAM is able to further improve the accuracy. This indicates the generality and applicability of proposed method across various long-tailed learning algorithms.

We show additional results on the large scale ImageNet-LT (Table 6) and iNaturalist 2018 (Table 3) dataset with LDAM-DRW. We also compare with recent long-tail learning methods: cRT [27], MisLAS [57], DisAlign [55] and DRO-LT [44]. On ImageNet-LT, LDAM+DRW with SAM leads to a 3.2% gain in overall accuracy with 6.5% increase in tail class accuracy. It can be seen that LDAM+DRW+SAM outperforms most other methods, including MisLAS which uses mixup. Also, it is important to note that MisLAS is trained for 180 epochs unlike LDAM+DRW which is trained only for 90 epochs. We observe that LDAM+DRW+SAM surpasses the performance of two-stage training methods including MisLAS, cRT, LWS, and DisAlign. Compared to these two-stage methods, our method is a single stage method and outperforms these two-stage methods. We want to add that

[4] https://github.com/pseulki/IB-Loss

[5] https://github.com/dvlab-research/Parametric-Contrastive-Learning

Table 7: Results on CIFAR-10 LT with different Imbalance Factor (β).

	β = 10				β = 50			
	Acc	Head	Mid	Tail	Acc	Head	Mid	Tail
CE + DRW [8]	88.3	93.6	85.3	86.9	79.9	92.2	76.5	72.0
CE + DRW + SAM	89.7	93.4	86.1	90.8	83.8	91.3	80.5	80.8
LDAM + DRW [8]	87.8	91.9	85.0	87.5	82.0	90.9	78.7	77.5
LDAM + DRW + SAM	89.4	93.4	86.2	89.8	84.8	92.8	82.1	80.4
	β = 100				β = 200			
	Acc	Head	Mid	Tail	Acc	Head	Mid	Tail
CE + DRW [8]	75.5	91.6	74.1	61.4	69.9	91.1	70.0	48.4
CE + DRW + SAM	80.6	91.4	78.0	73.1	76.6	91.5	74.9	64.0
LDAM + DRW [8]	77.5	91.1	75.7	66.4	72.5	90.2	72.3	54.9
LDAM + DRW + SAM	81.9	91.0	79.2	76.4	78.1	91.2	75.6	68.4

we were not able to reproduce the numbers reported in DRO-LT* [44] when we were trying to incorporate SAM with DRO-LT.

With LDAM+DRW, the addition of SAM results in an increase in Head, Mid and Tail categories on iNaturalist 2018 (Table 3). Specifically, LDAM+DRW+SAM outperforms all other methods in the tail class accuracy.

This further emphasizes that our analysis is applicable to large scale imbalanced datasets like ImageNet-LT and iNaturalist 2018. We also want to highlight that our analysis shows that high ρ SAM with re-weighting can be used as a *strong baseline* in long tailed visual recognition problem. We also find that SAM is highly compatible with different loss-based methods (like LDAM, VS) for tackling imbalance and can be used to achieve significantly better performance.

F Additional Results with Varying Imbalance Factor

We show the results with different imbalance factors (β = 10, 50, 100 and 200) on CIFAR-10 LT (Table 7) and CIFAR-100 LT (Table 8) datasets with two methods. It can be seen that the observations in Table 1 are applicable with different degrees of imbalance. SAM with re-weighting improves upon the performance of CE and LDAM losses in all the experiments with varied imbalance factor. We observe an average increase of 3.9% and 3.2% on CIFAR-10 LT and CIFAR-100 LT datasets, respectively. This gain in performance is primarily due to the improvement in the tail accuracy, which increases by 8.6% on CIFAR-10 LT and 3.9% on CIFAR-100 LT.

As the dataset becomes more imbalanced (β increases), the gain in accuracy with SAM on the tail classes improves significantly. For instance, on CIFAR-10 LT with β = 10 (Table 7), CE+DRW+SAM improves upon CE+DRW by 1.2% with a 3.9% increase in tail class accuracy. However, with a more imbalanced dataset (*i.e.* CIFAR-10 LT β = 200), SAM leads to a 6.7% boost in overall accuracy with a massive 15.6% increase in the tail class performance.

G Algorithm

We describe our method in detail in Algorithm 1. On the large scale ImageNet-LT and iNaturalist-18 dataset, we use $\rho_{drw} > \rho$. For CIFAR-10 LT and CIFAR-100 LT, we find that $\rho = \rho_{drw}$ works well.

H Related Work: Long-tailed Learning

In this section, we discuss some recent approaches in long-tailed learning. Equalization loss is proposed in Tan et al. [45] based on the proposition that the gradients of negative samples overpower

Table 8: Results on CIFAR-100 LT with different Imbalance Factor (β).

	$\beta = 10$				$\beta = 50$			
	Acc	Head	Mid	Tail	Acc	Head	Mid	Tail
CE + DRW [8]	58.1	65.6	58.5	48.2	46.5	63.3	47.5	24.4
CE + DRW + SAM	60.7	66.0	60.5	54.4	50.0	61.9	50.9	33.7
LDAM + DRW [8]	57.8	67.5	58.9	44.5	47.1	62.9	48.2	26.1
LDAM + DRW + SAM	60.1	70.2	61.3	46.1	49.4	66.1	50.2	27.8
	$\beta = 100$				$\beta = 200$			
	Acc	Head	Mid	Tail	Acc	Head	Mid	Tail
CE + DRW [8]	41.0	61.3	41.7	14.7	36.9	59.7	36.1	9.6
CE + DRW + SAM	44.6	61.2	47.5	20.7	41.7	63.4	43.0	13.1
LDAM + DRW [8]	42.7	61.8	42.2	19.4	38.3	58.8	36.3	15.1
LDAM + DRW + SAM	45.4	64.4	46.2	20.8	42.0	63.0	41.4	16.6

Table 9: Results on CIFAR-100 LT with IB-Loss and PaCo.

	$\beta = 100$		$\beta = 200$	
	Acc	Tail	Acc	Tail
IB [8]	40.4	14.9	36.7	10.3
IB + SAM	42.8	25.0	37.7	17.8
PaCo [13]	51.5	33.9	47.0	26.9
PaCo + SAM	53.0	36.0	48.0	27.8

the gradient of positive samples for minority classes. Influence-Balanced Loss [40] is a sample-level re-weighting method that reweights each sample by the inverse of the norm of the gradient of each sample. The gradient of each sample estimates the influence of that sample in determining the decision boundary. Distill the Virtual Examples (DiVE) [23] addresses the problem of class-imbalanced learning from the lens of knowledge distillation. It is shown that the teacher models' predictions (virtual examples) can be distilled into the student model by making use of cross-category interactions. This leads to an improvement in the accuracy of the minority class samples.

Self-Supervised Learning methods have been shown to learn generalizable representations [11] which are useful for a wide variety of downstream tasks. Self-Supervised pre-training (SSP) has been shown to improve the performance of class-imbalanced learning [50]. Parametric Contrastive Learning (PaCo) [13] introduces parametric class-wise learnable centers into the Supervised Contrastive Learning [29] framework to improve the performance on imbalanced datasets. PaCo achieves close to state-of-the-art performance on most of the long-tailed learning benchmarks. Self Supervised to Distillation (SSD) [34] is a multi-stage training framework for long-tailed recognition with a total of four stages of training. The first two stages involve self-supervised training followed by the generation of soft labels. The final two stages include joint training with distillation and classifier fine-tuning. Balanced Contrastive Learning (BCL) [58] adapts the Supervised Contrastive framework [29] by proposing a Balanced Contrastive loss which ensures that the feature space is balanced when training with an imbalanced dataset.

I Code and License Details

Our codebase is derived from the official implementation of LDAM-DRW[8][6], VS-Loss [30][7] and SAM[17][8] which have been released under the MIT license. We have included the code and the

[6] https://github.com/kaidic/LDAM-DRW
[7] https://github.com/orparask/VS-Loss
[8] https://github.com/davda54/sam

Algorithm 1 DRW + SAM

Require: Network g with parameters w; Training set $\mathbb{S}$; Batch size b; Learning rate $\eta > 0$; Neighborhood size $\rho > 0$, Neighborhood size for re-weighted loss $\rho_{drw} >= \rho$; Total Number of Iterations E; Deferred Reweighting Threshold T; Number of samples in class y: n_y; Loss Function $\mathcal{L}$ (Cross-Entropy, LDAM).

for $i = 1$ to E **do**
  Sample a mini-batch $\mathbb{B} \subset \mathbb{S}$ with size b.
  if $E < T$ **then**
    Compute Loss $\mathcal{L} \leftarrow \frac{1}{b} \sum_{(x,y)\in\mathbb{B}} \mathcal{L}(y; g_w(x))$
    Compute $\epsilon \leftarrow \rho * \nabla_w \mathcal{L} / ||\nabla_w \mathcal{L}||$ ▷ Compute Sharp-Maximal Point
    Compute Loss at $w + \epsilon$; $\mathcal{L} \leftarrow \frac{1}{b} \sum_{(x,y)\in\mathbb{B}} \mathcal{L}(y; g_{w+\epsilon}(x))$
    Calculate gradient d: $d \leftarrow \nabla_w \mathcal{L}$
  else ▷ Deferred Re-Weighting (DRW)
    Compute re-weighted Loss $\mathcal{L}_{\text{RW}} \leftarrow \frac{1}{b} \sum_{(x,y)\in\mathbb{B}} n_y^{-1} \cdot \mathcal{L}(y; g_w(x))$
    Compute $\epsilon \leftarrow \rho_{drw} * \nabla_w \mathcal{L}_{\text{RW}} / ||\nabla_w \mathcal{L}_{\text{RW}}||$
    Compute re-weighted Loss at $w + \epsilon$; $\mathcal{L}_{\text{RW}} \leftarrow \frac{1}{b} \sum_{(x,y)\in\mathbb{B}} n_y^{-1} \cdot \mathcal{L}(y; g_{w+\epsilon}(x))$
    Calculate gradient d: $d \leftarrow \nabla_w \mathcal{L}_{\text{RW}}$
  Update weights $w_{i+1} \leftarrow w_i - \eta d$

pretrained weights of the CE+DRW model trained of CIFAR-10 LT in the supplementary material. The code to reproduce the experiments is available at https://github.com/val-iisc/Saddle-LongTail.

References

[1] Guillaume Alain, Nicolas Le Roux, and Pierre-Antoine Manzagol. Negative eigenvalues of the hessian in deep neural networks. *arXiv preprint arXiv:1902.02366*, 2019. 4

[2] Maksym Andriushchenko and Nicolas Flammarion. Understanding sharpness-aware minimization, 2022. URL https://openreview.net/forum?id=qXa0nhTRZGV. 6

[3] Raman Arora, SANJEEV Arora, Joan Bruna, NADAV Cohen, SIMON DU, RONG GE, SURIYA GUNASEKAR, C Jin, JASON LEE, TENGYU MA, et al. Theory of deep learning, 2020. 3

[4] Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com. 14

[5] Devansh Bisla, Jing Wang, and Anna Choromanska. Low-pass filtering sgd for recovering flat optima in the deep learning optimization landscape. *arXiv preprint arXiv:2201.08025*, 2022. 4, 10, 14

[6] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018. 1, 3

[7] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International Conference on Machine Learning*, pages 872–881. PMLR, 2019. 1

[8] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019. 1, 3, 7, 8, 13, 17, 18

[9] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12): 124018, 2019. 2, 4

[10] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16: 321–357, 2002. 3

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 18

[12] Andrew Cotter, Maya Gupta, Heinrich Jiang, Nathan Srebro, Karthik Sridharan, Serena Wang, Blake Woodworth, and Seungil You. Training well-generalizing classifiers for fairness metrics and other data-dependent constraints. In *International Conference on Machine Learning*, pages 1397–1405. PMLR, 2019. 1

[13] Jiequan Cui, Zhisheng Zhong, Shu Liu, Bei Yu, and Jiaya Jia. Parametric contrastive learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 715–724, 2021. 9, 16, 18

[14] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 1

[15] Hadi Daneshmand, Jonas Kohler, Aurelien Lucchi, and Thomas Hofmann. Escaping saddles with stochastic gradients. In *International Conference on Machine Learning*, pages 1155–1164. PMLR, 2018. 2, 4, 6, 7, 12

[16] Yann N Dauphin, Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, Surya Ganguli, and Yoshua Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in neural information processing systems*, 27, 2014. 2, 4, 5

[17] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=6Tm1mposlrM. 4, 14, 18

[18] Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In *Conference on learning theory*, pages 797–842. PMLR, 2015. 2, 4

[19] Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via hessian eigenvalue density. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2232–2241. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ghorbani19b.html. 2, 4, 5, 14

[20] Justin Gilmer, Behrooz Ghorbani, Ankush Garg, Sneha Kudugunta, Behnam Neyshabur, David Cardoze, George Dahl, Zachary Nado, and Orhan Firat. A loss curvature perspective on training instability in deep learning. *arXiv preprint arXiv:2110.04369*, 2021. 5, 14

[21] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. doi: 10.1109/TKDE.2008.239. 3

[22] 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 (CVPR)*, June 2016. 5

[23] Yin-Yin He, Jianxin Wu, and Xiu-Shen Wei. Distilling virtual examples for long-tailed recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 235–244, October 2021. 18

[24] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1):1–42, 1997. 4

[25] Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International Conference on Machine Learning*, pages 1724–1732. PMLR, 2017. 2, 4, 10, 14

[26] Chi Jin, Praneeth Netrapalli, Rong Ge, Sham M. Kakade, and Michael I. Jordan. Stochastic gradient descent escapes saddle points efficiently. *ArXiv*, abs/1902.04811, 2019. 2, 4, 14

[27] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1gRTCVFvB. 3, 9, 14, 16

[28] 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. 4

[29] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 18

[30] Ganesh Ramachandra Kini, Orestis Paraskevas, Samet Oymak, and Christos Thrampoulidis. Label-imbalanced and group-sensitive classification under overparameterization. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 3, 7, 8, 9, 18

[31] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1):32–73, 2017. 1

[32] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[33] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018. 5

[34] Tianhao Li, Limin Wang, and Gangshan Wu. Self supervision to distillation for long-tailed visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 630–639, 2021. 18

[35] Xinyan Li, Qilong Gu, Yingxue Zhou, Tiancong Chen, and Arindam Banerjee. Hessian based analysis of sgd for deep nets: Dynamics and generalization. In *Proceedings of the 2020 SIAM International Conference on Data Mining*, pages 190–198. SIAM, 2020. 5

[36] Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. *arXiv preprint arXiv:2203.02714*, 2022. 6

[37] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2537–2546, 2019. 8

[38] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020. 3

[39] DM Merkulov and Ivan V Oseledets. Empirical study of extreme overfitting points of neural networks. *Journal of Communications Technology and Electronics*, 64(12):1527–1534, 2019. 5

[40] Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi. Influence-balanced loss for imbalanced visual classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 735–744, October 2021. 9, 16, 18

[41] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf. 14

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 1, 8

[43] Levent Sagun, Utku Evci, V Ugur Guney, Yann Dauphin, and Leon Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv preprint arXiv:1706.04454*, 2017. 4

[44] Dvir Samuel and Gal Chechik. Distributional robustness loss for long-tail learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9495–9504, October 2021. 3, 9, 14, 16, 17

[45] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, Changqing Yin, and Junjie Yan. Equalization loss for long-tailed object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11662–11671, 2020. 17

[46] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 1

[47] Grant Van Horn and Pietro Perona. The devil is in the tails: Fine-grained classification in the wild. *arXiv preprint arXiv:1709.01450*, 2017. 1

[48] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018. 8, 9

[49] Xudong Wang, Long Lian, Zhongqi Miao, Ziwei Liu, and Stella Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=D9I3drBz4UC. 3

[50] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19290–19301. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/e025b6279c1b88d3ec0eca6fcb6e6280-Paper.pdf. 18

[51] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. *Advances in Neural Information Processing Systems*, 33:19290–19301, 2020. 9

[52] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pages 581–590. IEEE, 2020. 14

[53] Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning. *arXiv preprint arXiv:2001.01385*, 2020. 3

[54] Chuanhai Zhang. *Medical image classification under class imbalance*. PhD thesis, Iowa State University, 2019. 1

[55] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2361–2370, 2021. 9, 14, 16

[56] Yifan Zhang, Bryan Hooi, Lanqing Hong, and Jiashi Feng. Test-agnostic long-tailed recognition by test-time aggregating diverse experts with self-supervision. *arXiv preprint arXiv:2107.09249*, 2021. 3, 9

[57] Shu Liu Zhisheng Zhong, Jiequan Cui and Jiaya Jia. Improving calibration for long-tailed recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3, 8, 9, 14, 16

[58] Jianggang Zhu, Zheng Wang, Jingjing Chen, Yi-Ping Phoebe Chen, and Yu-Gang Jiang. Balanced contrastive learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6908–6917, June 2022. 18

[59] Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, sekhar tatikonda, James s Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=edONMAnhLu-. 6