Appendix for: Class-Conditional Sharpness-Aware Minimization for Deep Long-Tailed Recognition

A. Micro Benchmarks

A.1. Additional Ablation Studies

We conduct additional ablation studies on CIFAR-10-LT and CIFAR-100-LT when the imbalance ratios are 50, 100, and 200, and the results are shown in Table 1, Table 2, and Table 3, respectively. The presented results still well-support the effectiveness of each stage operation.

Dataset	cRT	stage-1	stage-2	Acc.
CIFAR10-LT	✔			81.14
	✔	✔		84.03
	✔		✔	85.69
	✔	✔	✔	**85.81**
CIFAR100-LT	✔			47.73
	✔	✔		52.41
	✔		✔	52.73
	✔	✔	✔	**53.24**

Table 1. Ablation studies on CIFAR-10-LT and CIFAR-100-LT when the imbalance ratio is 50.

Dataset	cRT	stage-1	stage-2	Acc.
CIFAR10-LT	✔			75.12
	✔	✔		79.74
	✔		✔	81.68
	✔	✔	✔	**82.42**
CIFAR100-LT	✔			43.39
	✔	✔		47.74
	✔		✔	48.20
	✔	✔	✔	**48.77**

Table 2. Ablation studies on CIFAR-10-LT and CIFAR-100-LT when the imbalance ratio is 100.

A.2. Impact of the Perturbation Scale

We empirically examine the impact of the perturbation scale to further verify the effectiveness of our derived characteristic radius:

$$\rho_c^* = \left(\frac{||\boldsymbol{w}||_2}{2||\nabla_w L_{\mathcal{S}}^c(\boldsymbol{w})||_2}\right)^{\frac{1}{2}} k^{-\frac{1}{4}}(n_c-1)^{-\frac{1}{4}} \quad (1)$$

By multiplying a scaling factor $\boldsymbol{r}$ on the Eqn 1, we scale the perturbation norm from 10^{-1} to 10^{-7}, and the corresponding results are presented in Figure 1. From the results, we observe that CC-SAM performs best when $\boldsymbol{r}$ is set as 10^{-2}, which demonstrates that there indeed exists a optimal perturbation scale. However, the coefficient of the characteristic radius ρ^* (Eqn (4) in the main text) is not an exact number and should be empirically tuned (see more elaboration on this point in the **Remarks** of Section D).

Dataset	cRT	stage-1	stage-2	Acc.
CIFAR10-LT	✔			69.99
	✔	✔		76.51
	✔		✔	77.43
	✔	✔	✔	**78.31**
CIFAR100-LT	✔			37.51
	✔	✔		43.12
	✔		✔	44.07
	✔	✔	✔	**44.99**

Table 3. Ablation studies on CIFAR-10-LT and CIFAR-100-LT when the imbalance ratio is 200.

A.3. The Superiority and Versatility of CC-SAM

As an augment version in DLTR, we apply CC-SAM to LDAM and compare it with vanilla SAM in Table 4, from which we can observe that CC-SAM is more beneficial to LDAM than SAM, indicating its superiority in adapting to DLTR. Besides, to show the versatility of CC-SAM, we further equip it on GCL and show the improvement in Table 4.

A.4. Additional Flattening Operations Test on Places-LT

To support our motivation, we further conduct verification on Places-LT. By integrating model perturbation and spectral normalization into the first stage of training, we

CIFAR-10-LT	LDAM			GCL		
Imb.	200	100	50	200	100	50
Vanilla	70.15	73.26	78.16	79.03	82.68	85.46
+ SAM	68.59	74.20	78.88	-	-	-
+ CC-SAM	**71.05**	**74.56**	**79.34**	**80.74**	**83.54**	**86.07**

Table 4. Improvement of applying CC-SAM to LDAM and GCL.

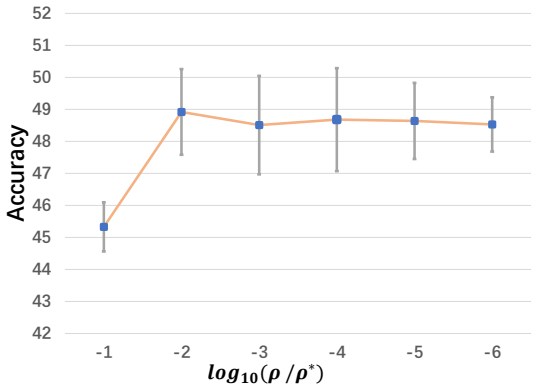


Figure 1. The impact of the perturbation scale.

show that cRT still cannot be well improved and still has much room compared to MisLAS and GCL.

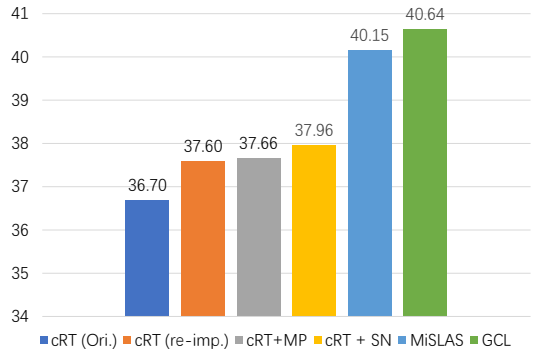


Figure 2. The results of simply applying flattening operations. 're-imp.' represents the re-implementation of cRT while 'Ori' is the report result from the original paper. 'cRT + MP' represents applying model perturbation to cRT while 'cRT + SN' represents integrating spectral normalization into cRT.

A.5. Training Time Cost

CC-SAM is intuitively more complex than its baselines, as it requires multiple forward and an additional backward computing. But since we only perturb a part of the model parameters in our implementation and such an operation is realized in a mini-batch, which contains fewer classes, it results in a lower time cost than expected. Here we provide a simple comparison of the time cost:

Table 5. Training time cost comparison running on Tesla T4.

CE	LDAM	CC-SAM
0.62h	0.68h	0.96h

B. Empirical Verification of the Nonvacuous Bound

To verify that the error bound in Theorem 1 provides a meaningful proxy for generalization performance (i.e., non-vacuous), we empirically observe the value of training loss, testing loss, and error bound of the head class (the class with the greatest number of samples, 5000 on CIFAR-10-LT) on CIFAR-10-LT. The corresponding curves are presented in Figure 3. According to their changing values, we can find that the square root term of Eqn 16 falls below 1 most of the time.

C. Experimental Setting of Flattening Operations

C.1. SWA

SWA [4] seeks the flat minima by averaging the SGD trajectory. We use the PyTorch official implementation [1] of SWA to experiment on each long-tailed model. During each stage, the suggested standard decaying learning rate strategy is used for the first 75% of training time, while the learning rate is set to a high constant for the remaining 25% of the time, where weight averaging occurs.

C.2. Spectral Normalization

Spectral normalization [1] endows the model 1-Lipschitz continuity by dividing the square root of the largest eigenvalue. To release the computation burden, we adopt the

[1] https://pytorch.org/blog/stochastic-weight-averaging-in-pytorch/.

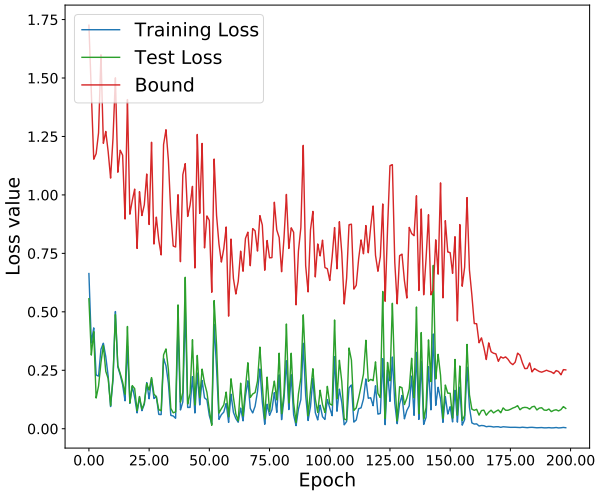


Figure 3. Training loss, testing loss, and error bound of CC-SAM vs. training epochs. All curves are evaluated on the full training/test sets using the same set of model parameters at the end of each epoch, and are plotted for the whole period of stage 1 (200 epochs).

power iteration method [2] to estimate the eigenvalue, and the iteration is set to 5.

C.3. Model Perturbation

Following the recent paper [8], we perturb the model parameters by adding Gaussian noise with a similar scale as our method. Specifically, here we only perturb the classifier with a Gaussian noise, whose norm is 0.001.

C.4. Gradient Normalization

As an advanced version of SAM, gradient normalization [9] penalizes the gradient via an approximation of the Hassian matrix:

$$\nabla_{\boldsymbol{\omega}} L(\boldsymbol{\omega}) = (1-\alpha)\nabla_{\boldsymbol{\omega}} L_S(\boldsymbol{\omega}) + \alpha * \nabla_{\boldsymbol{\omega}} L_S(\boldsymbol{\omega} + r * \frac{\nabla_{\boldsymbol{\omega}} L_S(\boldsymbol{\omega})}{\|\nabla_{\boldsymbol{\omega}} L_S(\boldsymbol{\omega})\|}) \tag{2}$$

where α and r are hyper-parameters. This method derives the same perturbation direction as CC-SAM, but it still works for i.i.d. situations while lacking the long-tailed specific designs. We re-implement and integrate it into long-tailed models according to its publicly released code[3].

D. Proof of Theorem 1 (Perturbative PAC-Bayesian Generalization Bound)

Following the convention of the main text, we denote scalars as s, vectors as $\boldsymbol{s}$, matrices as $\boldsymbol{S}$, sets as $\mathcal{S}$, the number of samples as n and the number of parameters as k. To apply the PAC-Bayesian framework [7] to the generalization of deep neural networks, we follow [2]. For any prior P and posterior Q over the parameters $\boldsymbol{w}$ with probability at least $1-\delta$, over the choice of the training set $\mathcal{S}$, the following generalization bound holds:

$$\mathbb{E}_{\boldsymbol{w}\sim Q}[L_{\mathcal{T}}(\boldsymbol{w})] \leq \mathbb{E}_{\boldsymbol{w}\sim Q}[L_{\mathcal{S}}(\boldsymbol{w})] + \sqrt{\frac{KL(Q||P) + \log\frac{n}{\delta}}{2(n-1)}}. \tag{3}$$

We further adopt the condition

$$L_{\mathcal{T}}(\boldsymbol{w}) \leq \mathbb{E}_{\boldsymbol{w}\sim Q}[L_{\mathcal{T}}(\boldsymbol{w})] \tag{4}$$

from [3], meaning that adding Gaussian perturbation should not decrease the test error. Note that this is expected to hold in practice for the final solution but does not necessarily hold for any $\boldsymbol{w}$.

The KL-divergence between two k-dimensional Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_P, \boldsymbol{\Sigma}_P)$ and $\mathcal{N}(\boldsymbol{\mu}_Q, \boldsymbol{\Sigma}_Q)$ is given by:

$$KL(\mathcal{N}(\boldsymbol{\mu}_P, \boldsymbol{\Sigma}_P)||\mathcal{N}(\boldsymbol{\mu}_Q, \boldsymbol{\Sigma}_Q)) = \tag{5}$$

$$\frac{1}{2}\left[\mathrm{tr}(\boldsymbol{\Sigma}_Q^{-1}\boldsymbol{\Sigma}_P) + (\boldsymbol{\mu}_Q - \boldsymbol{\mu}_P)^T\boldsymbol{\Sigma}_Q^{-1}(\boldsymbol{\mu}_Q - \boldsymbol{\mu}_P) - k + \log\left(\frac{\det\boldsymbol{\Sigma}_Q}{\det\boldsymbol{\Sigma}_P}\right)\right]. \tag{6}$$

To simplify Eqn 6, let the prior P and the posterior Q be a k-dimensional isotropic Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_P, \sigma_P^2\boldsymbol{I})$ and $\mathcal{N}(\boldsymbol{\mu}_Q, \sigma_Q^2\boldsymbol{I})$ respectively, the KL divergence can be written as:

$$KL(Q||P) = \frac{1}{2}\left[\frac{k\sigma_Q^2 + ||\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q||_2^2}{\sigma_P^2} - k + k\log\left(\frac{\sigma_P^2}{\sigma_Q^2}\right)\right]. \tag{7}$$

Given $\boldsymbol{\mu}_P, \boldsymbol{\mu}_Q$ and σ_Q, the KL divergence can be minimized by an optimal ${\sigma_P^*}^2 = \sigma_Q^2 + ||\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q||_2^2/k$, yielding

$$KL(Q||P) \leq \frac{k}{2}\log\left(1 + \frac{||\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q||_2^2}{k\sigma_Q^2}\right). \tag{8}$$

Let $\sigma_Q = \sigma$, $\boldsymbol{\mu}_Q = \boldsymbol{w}$ and use the reparameterization trick that $\boldsymbol{w} \leftarrow \boldsymbol{w} + \boldsymbol{\epsilon}$, assuming $\boldsymbol{\mu}_P = \boldsymbol{0}$ (parameters are initialized by a zero-mean Gaussian prior), the generalization bound in Eqn 3 becomes

$$\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\sigma^2\boldsymbol{I})}[L_{\mathcal{T}}(\boldsymbol{w})] \leq \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\sigma^2\boldsymbol{I})}[L_{\mathcal{S}}(\boldsymbol{w})] + \tag{9}$$

$$\sqrt{\frac{\frac{k}{4}\log\left(1 + \frac{||\boldsymbol{w}||_2^2}{k\sigma^2}\right) + \frac{1}{2}\log\frac{n}{\delta}}{(n-1)}}. \tag{10}$$

[2] https://en.wikipedia.org/wiki/Power_iteration
[3] https://github.com/zhaoyang-0204/gnp

Since $\log(1+x) < x$ for all $x > 0$, providing a very tight bound in the over-parameterized regime ($k \gg \frac{||\boldsymbol{w}||_2^2}{\sigma^2}$, which generally holds for modern neural networks, see experiments in [2]), we have

$$\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}[L_{\mathcal{T}}(\boldsymbol{w})] \leq \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}[L_{\mathcal{S}}(\boldsymbol{w})] + \sqrt{\frac{\frac{||\boldsymbol{w}||_2^2}{4\sigma^2} + \frac{1}{2}\log\frac{n}{\delta}}{(n-1)}}. \tag{11}$$

In the above bound, $||\boldsymbol{\epsilon}||_2^2$ has chi-square distribution and by Lemma 1 in [6], we have that for any positive t:

$$P(||\boldsymbol{\epsilon}||_2^2 - k\sigma^2 \geq 2\sigma^2\sqrt{kt} + 2t\sigma^2) \leq \exp(-t). \tag{12}$$

Therefore, with probability at least $1 - 1/\sqrt{n}$ we have that

$$\begin{aligned} ||\boldsymbol{\epsilon}||_2^2 &\leq \sigma^2\left(2\ln(\sqrt{n}) + k + 2\sqrt{k\ln(\sqrt{n})}\right) \\ &\leq \sigma^2 k\left(1 + \sqrt{\frac{\ln(n)}{k}}\right)^2 = k\rho^2. \end{aligned} \tag{13}$$

Substituting the above value for ρ back to the inequality and using the assumption 4 and the training loss $L_{\mathcal{S}}(\boldsymbol{w}) \leq 1$, gives us following inequality:

$$L_{\mathcal{T}}(\boldsymbol{w}) \leq \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma^2\boldsymbol{I})}[L_{\mathcal{T}}(\boldsymbol{w})] \quad \text{(Eqn 4)} \tag{14}$$

$$\leq (1 - 1/\sqrt{n}) \max_{||\boldsymbol{\epsilon}||_2 \leq \sqrt{k}\rho} L_{\mathcal{S}}(\boldsymbol{w}+\boldsymbol{\epsilon}) + 1/\sqrt{n} + \sqrt{\frac{\frac{||\boldsymbol{w}||_2^2}{4\rho^2}\left(1+\sqrt{\frac{\log(n)}{k}}\right)^2 + \frac{1}{2}\log\frac{n}{\delta}}{(n-1)}} \tag{15}$$

$$\leq \max_{||\boldsymbol{\epsilon}||_2 \leq \sqrt{k}\rho} L_{\mathcal{S}}(\boldsymbol{w}+\boldsymbol{\epsilon}) + \sqrt{\frac{\frac{||\boldsymbol{w}||_2^2}{4\rho^2} + \log\frac{n}{\delta} + \mathcal{O}(1)}{(n-1)}}, \tag{16}$$

which is exactly Theorem 1, where we arguably assume that $\log(n), \sqrt{k\log(n)} \ll \frac{||\boldsymbol{w}||_2^2}{\rho^2} < n, k$.

Remarks: Note that the assumption that the optimal ${\sigma_P^*}^2 = \sigma_Q^2 + ||\boldsymbol{\mu}_P - \boldsymbol{\mu}_Q||_2^2/k$ can be attained might be unrealistic since σ_P should be chosen before observing the training data $\mathcal{S}$, whereas $\boldsymbol{\mu}_Q$, σ_Q generally depend on $\mathcal{S}$. However, a relaxed bound can be derived by having a set of predefined values for σ_P and pick the best one in that set. See [3] and Theorem 2 of [5] for the discussion around this technique, which results in an $\mathcal{O}(1) = 1 + 8\log(6n + 3k)$. Moreover, the approximations and the lesser terms (e.g. $1/\sqrt{n}$ in Eqn 15 and $\mathcal{O}(1)$ in Eqn 16) we made in the process of deriving Theorem 1 indicates that *the constant coefficient of the characteristic radius ρ^* (Eqn (4) in the main text) is not an exact number, which should be empirically tuned to realize the full potential of CC-SAM, and is also beneficial for accommodating tighter/looser versions of PAC-Bayesian bound*. However, the correlation between ρ^* and the label frequency n as well as the gradient $\nabla_{\boldsymbol{w}} L_{\mathcal{S}}(\boldsymbol{w})$ do shed light on the design choices of CC-SAM, which are proven effective in practice.

E. Loss Landscape

To better understand the sharpness of the deep long-tailed model, we visualize their 2D loss landscapes via a public tool [4] after training on CIFAR-10-LT in Figure 4-9. The horizontal coordinate and vertical coordinate are the step distances from the trained weights according to two random directions. The maximum distance is set as the 1/3 norm of trained weights. Taking CE as the baseline, we have the following observations, which are consistent with those in the main text:

- **LDAM** is definitely sharper than CE.
- **M2m** has a flatter region under the overall and tail evaluations but fails to be flat for head classes. This is because M2m adopts an adversarial example generation method to translate samples from head classes to tail classes for augmentation. Such a translation would intuitively improve the performance of tail classes while sacrificing the performance of head classes.
- **MisLAS** is still flatter than CE from each view.
- **GCL**'s loss landscapes looks pretty steep than others. Generally, it looks flatter than CE (when encountering large noise perturbations), but its local region (small noise perturbations) is sharper from each view, which has been concluded previously.
- **CC-SAM** is shown to have a rather flat minima. Moreover, it mainly presents an asymmetric valley as suggested in [4].

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2

[2] Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence*, 2017. 3, 4

[4] https://github.com/marcellodebernardi/loss-landscapes

[3] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *9th International Conference on Learning Representations*, 2021. 3, 4

[4] Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. *Advances in neural information processing systems*, 32, 2019. 2, 4

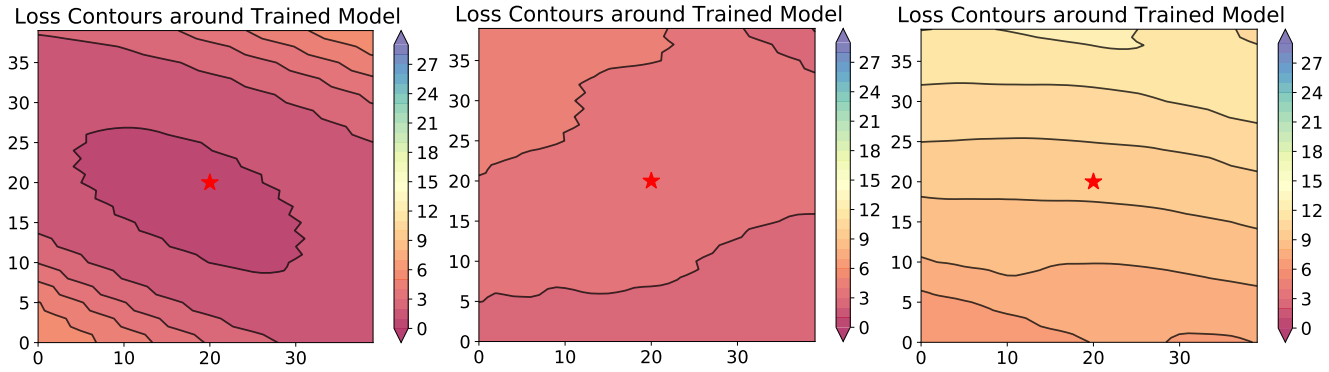
[5] John Langford and Rich Caruana. (not) bounding the true error. *Advances in Neural Information Processing Systems*, 14, 2001. 4

[6] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000. 4

[7] David A McAllester. Pac-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory*, pages 164–170, 1999. 3

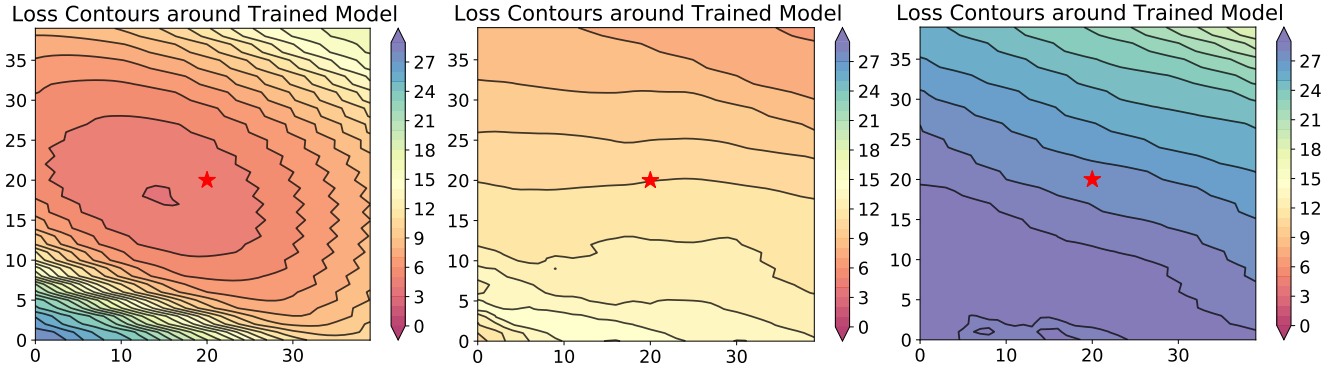
[8] Guangyuan Shi, Jiaxin Chen, Wenlong Zhang, Li-Ming Zhan, and Xiao-Ming Wu. Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. *Advances in Neural Information Processing Systems*, 34:6747–6761, 2021. 3

[9] Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. *arXiv preprint arXiv:2202.03599*, 2022. 3



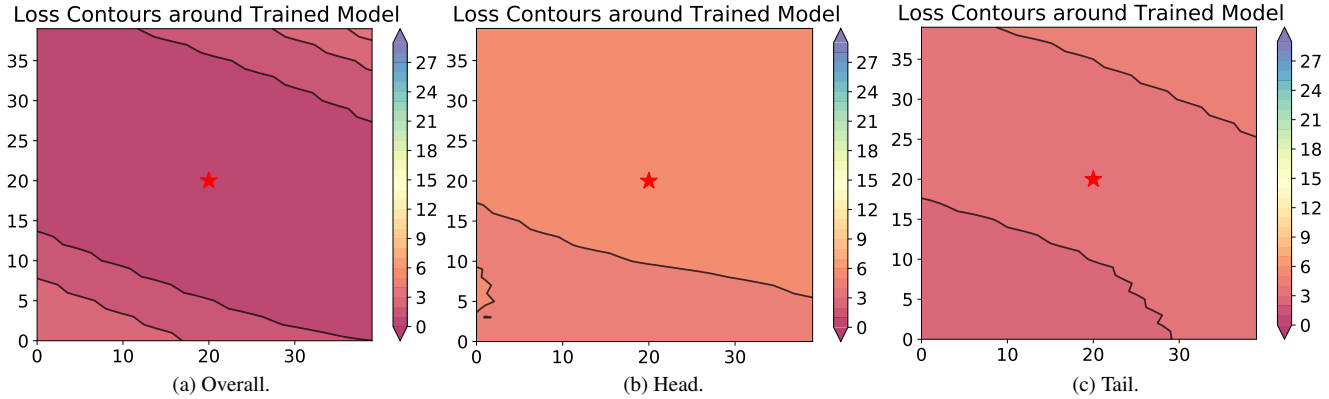
(a) Overall. (b) Head. (c) Tail.

Figure 4. Loss landscape of CE. The left figure is the overall loss landscape, which involves all classes, while the central figure shows that of the head class and the right figure shows that of the tail class. The red pentagram represents the point of the trained weights.



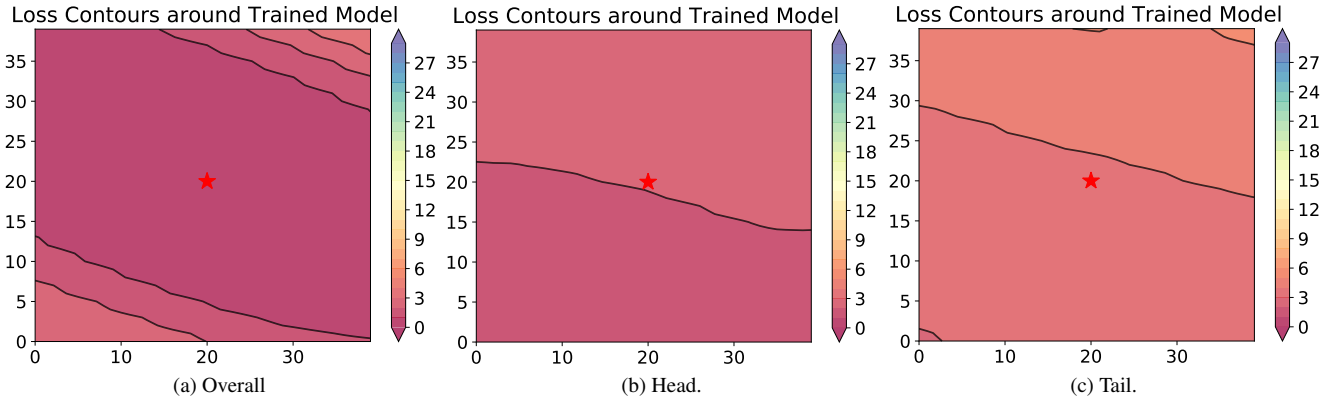
(a) Overall. (b) Head. (c) Tail.

Figure 5. Loss landscape of LDAM.



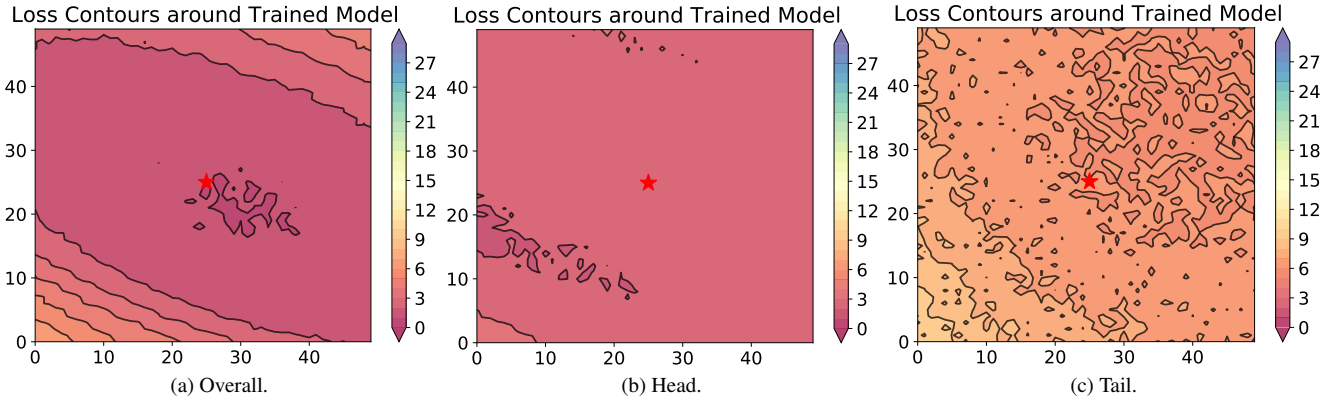
(a) Overall. (b) Head. (c) Tail.

Figure 6. Loss landscape of M2m.



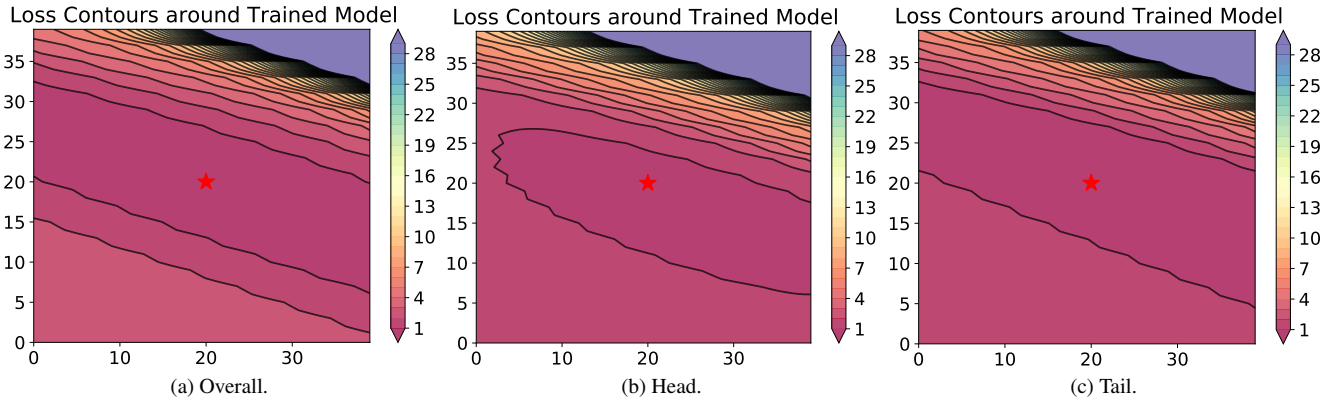
(a) Overall (b) Head. (c) Tail.

Figure 7. Loss landscape of MisLAS.



(a) Overall. (b) Head. (c) Tail.

Figure 8. Loss landscape of GCL.



(a) Overall. (b) Head. (c) Tail.

Figure 9. Loss landscape of CC-SAM.