Adversarial Robustness under Long-Tailed Distribution Supplementary Material

Tong Wu[1,5], Ziwei Liu[2], Qingqiu Huang[3], Yu Wang[4], Dahua Lin[1,5,6]
[1]The Chinese University of Hong Kong, [2]S-Lab, Nanyang Technological University, [3]Huawei,
[4]Tsinghua University, [5]SenseTime-CUHK Joint Lab, [6]Centre of Perceptual and Interactive Intelligence
{wt020,dhlin,hq016}@ie.cuhk.edu.hk, ziwei.liu@ntu.edu.sg, yu-wang@mail.tsinghua.edu.cn

A. Implementation Details of Experiments

A.1. Training Details and Hyper-parameter Setting

We adopt the WideResNet-34-10 as the model architecture. The initial learning rate is set as 0.1 with a decay factor of 10 at 60 and 75 epochs, totally 80 epochs. We use the last epoch for evaluation without early-stop for all the methods. We use the SGD momentum optimizer with weight decay set as 2×10^{-4}. We use a batch size of 64 for all the experiments in the main paper. The adversarial training is applied with the maximal permutation of $8/255$ and a step size of $2/255$ (0.031 and 0.0078 are used for implementation). The number of iterations in the inner maximization is set as 5, and a study on the effect of PGD steps in AT is reported in Sec. B.2. There are multiple hyper-parameters involved, where those that control margins or boundary adjustment are the most critical. Specifically, we adopt $m_0 = 0.1$ for CIFAR-10-LT and $m_0 \in \{0.2, 0.3\}$ for CIFAR-100-LT for different emphasis (i.e., the trade-off between natural and robust accuracy). $\tau_b - \tau_m = 1.2$ in Eqn.10 would basically produce a good result via training stage re-balancing, while $\tau_b - \tau_m = 0$ with $\tau_p = 1.5$ would also work well based on pure boundary adjustment at inference time. The optimal value of τ_p relies mainly on $\tau_b - \tau_m$. The ablation study includes detailed comparisons. Other hyper-parameters are less sensitive and have relatively small impact on the performance, where we adopt $s = 10$, $\gamma \in \{1/32, 1/16\}$, and we set $\alpha = 6, 3$ in Eqn.12 for CIFAR-10-LT and CIFAR-100-LT, respectively.

A.2. Code References

For the defense methods we compare with, we leverage the officially released code for them if available, including TRADES [23] [1], MMA [5] [2], Free [17] [3], and HE [15] [4]. AVmixup [10] are re-implement according to the paper.

For the attacks used for evaluation, we refer to several officially released code bases and the original papers for the implementation, including FGSM [7], PGD [13], MIM [6], C&W [2], and Auto Attack [3] [5].

For the long-tailed recognition methods in Table 1, we also refer to the official code of them if available.

A.3. Implementation Details of Table 1

In Sec.3.2 of the paper, we revisit and formulate a number of long-tailed recognition methods. We would report the hyper-parameters selected for them when combining with adversarial training framework in our implementation in Table 1, where we choose the **optimal values** by searching the hyper-parameters with a step size of 1 or 0.1.

B. Extensive Experiments

B.1. Loss Functions in Adversarial Training

In Sec.3, a modified loss function $\mathcal{L}'_{CE}$ can be adopted to AT procedure in three modes: replacing the CE in $\mathcal{L}_A$, $\mathcal{L}_T$, or both of them. We study the effect of the three modes in Table S2. It can be observed that: 1) replacing CE in $\mathcal{L}_A$ of the inner maximization would slightly benefit the natural accuracy with re-weighting [4], class-aware temperature [22], and bias [14, 16], while re-weighting would hurt robustness in this scenario; class-aware margin [1] is beneficial to robust accuracy but hurts the natural accuracy slightly; 2) replacing CE in $\mathcal{L}_T$ of the outer minimization or both $\mathcal{L}_A$ and $\mathcal{L}_T$ would result in a significantly higher natural accuracy with class-aware temperature and bias, and the robust accuracy also raises to some extent.

B.2. Effect of PGD Steps during Training

We use an iteration number of 5 with the step size set as $2/255$, approximately 0.0078, for the adversarial training

[1] https://github.com/yaodongyu/TRADES
[2] https://github.com/BorealisAI/mma_training
[3] https://github.com/mahyarnajibi/FreeAdversarialTraining
[4] https://github.com/ShawnXYang/AT_HE
[5] https://github.com/fra31/auto-attack

Table S1. Hyper-parameters selected for LT methods used in Table 1, where we choose the **optimal values** by searching the hyper-parameters with a step of 1 or 0.1. * denotes that we use CB-Focal.

Stage	Methods	Formulation	Hyper-parameters
Training	Vanilla FC	$g_i = W_i^T f(x)$	-
	Vanilla Cos	$g_i = \widetilde{W}_i^T \widetilde{f}(x)$	*temperature* $s = 16$
	Class-aware margin [1]	$g_i = W_i^T f(x) - \mathbb{1}\{i = y\} \cdot \delta_i$	$\delta_{max} = 0.5, \delta \propto n^{-1/4}$
	Cosine with margin [20, 15]	$g_i = \widetilde{W}_i^T \widetilde{f}(x) - \mathbb{1}\{i = y\} \cdot m$	$m = 0.2, s = 10$
	Class-aware temperature [22]	$g_i = W_i^T f(x) \cdot (n_i/n_{max})^\gamma$	$\gamma = 0.3$
	Class-aware bias [14, 16]	$g_i = {W_i}^T f(x) + \tau \log(n_i)$	$\tau = 1$
	Hard-example mining [11]	$r(y) = (1 - p_y)^\gamma$, applyed with BCE loss	$\gamma = 2$
	Re-sampling [18]	$r_s(i) \propto 1/n_i$	-
	Re-weighting* [4]	$r(y) = (1 - \beta)/(1 - \beta_y^n)$	$\beta = 0.9999, \gamma = 2$
Fine-tuning	One-epoch re-sampling [8]	$h_i = {W'}_i^T f(x)$, ${W'}_i$ re-trained with RS	-
	One-epoch re-weighting [1, 4]	$h_i = {W'}_i^T f(x)$, ${W'}_i$ fine-tuned with RW	$\beta = 0.9999, \gamma = 2$
	Learnable classifier scale [8]	$h_i = s_i \cdot W_i^T f(x)$, where s_i is learnable	-
Inference	Classifier re-scaling [22, 9]	$h_i = (W_i/n_i^\tau)^T f(x)$	$\tau = 0.3$
	Classifier normalization [8]	$h_i = (W_i/\|W_i\|^\tau)^T f(x)$	$\tau = 2$
	Class-aware bias [14]	$h_i = W_i^T f(x) - \tau \log(n_i)$	$\tau = 1$
	Feature disentangling [19]	$h_i = W_i^T (f(x) - \alpha \cos(f(x), d) \cdot d)$	$\alpha = 0.1$

Table S2. Different loss function applications in adversarial training. *Inner, outer*, or *both* denote to replace Cross-Entropy loss (CE) in the inner maximization of $\mathcal{L}_A$, outer minimization of $\mathcal{L}_T$, or both of them of Eqn.2 in the paper, respectively. A batch size of 128 is used here *different* from the main paper, which does not affect the relative comparison among them.

Method	Apply	Clean	PGD	AA
CE	both	62.29	28.14	26.78
Class-aware margin [1]	inner	61.27	28.22	28.23
	outer	60.70	28.04	26.75
	both	60.79	28.13	26.97
Re-weighting [4]	inner	66.77	22.15	21.07
	outer	62.76	32.76	27.77
	both	62.78	33.32	27.94
Class-aware temperature [22]	inner	63.98	26.89	25.96
	outer	72.93	30.71	29.45
	both	72.70	28.26	27.21
Class-aware bias [14, 16]	inner	64.09	27.27	27.31
	outer	71.33	29.25	27.82
	both	73.00	29.67	28.28

procedure. We adopt this setting for an acceptable balancing of natural and robust accuracy of the baseline. We study the effect of PGD iterations and step sizes in Table S3. As the iteration number increases, the natural accuracy is improved along with the decline of robust accuracy. Especially for CIFAR-10-LT that when we change from 5 steps to 7 steps, there is a sharp decrease in clean accuracy. As a result, we choose a 5-step PGD for the adversarial training framework in the paper.

Table S3. Effect of different iteration numbers and step size in the inner maximum of the adversarial training procedure.

Adversarial Training		CIFAR-10-LT		CIFAR-100-LT	
Iterations	Step size	Clean	PGD	Clean	PGD
1	0.031	64.94	25.39	47.96	14.23
3	0.010	64.03	26.44	47.33	15.32
5	0.0078	62.29	28.14	46.16	15.91
7	0.0078	58.92	29.70	45.23	16.82
10	0.0078	57.61	29.27	45.31	17.40

B.3. Intrinsic Properties among Classes

Apart from the distribution of sample numbers, different intrinsic properties and the confusion cross categories are also non-negligible factors that lead to varying performance among classes. As could be seen in Fig.1, when trained on balanced CIFAR-10, the difference in A_{nat} is relatively minor, while it reveals the disparity of difficulty and vulnerability among classes, leading to a significant variance in A_{rob}. Specifically, Class 2, 3, and 4 demonstrate signifi-

cantly lower robust accuracy compared with others.

To study this phenomenon, we train a network on the balanced CIFAR-10 and visualize the latent space via t-SNE in Fig. S1. It shows that the classes with lower A_{nat}, such as Class 3, obviously have less concentrated and partially overlapped distributions, making them easier to be attacked. It can also be observed that Class 2, 3, and 4 have clearly more dispersed distributions under the attack, which is consistent with their low A_{rob}. While under the long-tailed distribution, Class 3 benefits from the advantage of sample numbers over Class 4-9. Therefore, its accuracy becomes even higher than the original uniform distribution with the help of the induced prediction bias. A joint analysis of the effect by both intrinsic properties and the distribution of sample numbers among classes would be an interesting direction in the future.

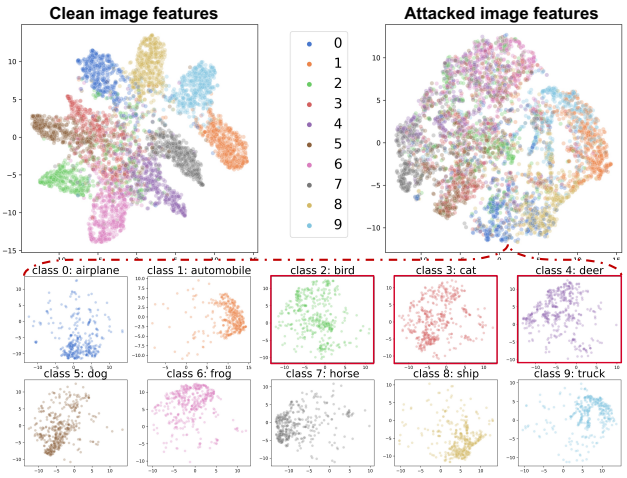

Figure S1. Latent space visualization before and after the attack.

B.4. Experiments on ImageNet-LT

We also evaluate our method on the more complicated ImageNet-LT [12] to encourage the exploration of real-world robustness. Due to the high resolution and large data scale, we adopt the standard single-step adversarial training (FGSM) and Fast adversarial training [21]. We use ResNet-50 as the backbone with $\epsilon = 2/255$ and $4/255$ following [17, 21]. The preliminary results are shown in Table S4.

Experimental results validate the effectiveness of our approach over the baseline. The relatively lower performance on ImageNet-LT compared to CIFAR also indicates that adversarial defense on the 1000-class ImageNet-LT is a more challenging problem, which is worth further exploration by the community.

C. Adversarial Attacks

Fast Gradient Sign Method (FGSM) [7] is a single-step attack that generates adversarial examples through a permutation along the gradient of the loss function with respect to the clean image as:

$$x^{adv} = x + \epsilon \cdot sign(\nabla_x \mathcal{L}_{CE}(x_t^{adv}, y)). \quad (1)$$

Table S4. Adversarial robustness results on ImageNet-LT.

Method	ϵ	CLEAN	FGSM	PGD-20
FAST-AT	2 / 255	11.36	8.23	7.16
FAST-Our		**15.45**	**11.51**	**10.31**
FGSM-AT	2 / 255	25.64	15.32	14.59
FGSM-Our		**30.02**	**18.50**	**17.67**
FAST-AT	4 / 255	7.20	4.52	3.76
FAST-Our		**10.76**	**7.28**	**6.13**
FGSM-AT	4 / 255	21.94	10.88	9.45
FGSM-Our		**25.88**	**13.49**	**11.87**

Projected Gradient Descent (PGD) [13] starts from an initialization point that is uniformly sampled from the allowed $\epsilon - ball$ centered at the clean image, and it extends FGSM by iteratively applying multiple small steps of permutation updating with respect to the current gradient as:

$$x_{t+1}^{adv} = clip_{x,\epsilon}(x_t^{adv} + \eta \cdot sign(\nabla_x \mathcal{L}_{CE}(x_t^{adv}, y))). \quad (2)$$

Momentum Iterative gradient-based Methods (MIM) [6] integrates the momentum into BIM with a decay factor μ,

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x \mathcal{L}_{CE}(x_t^{adv}, y)}{\left\|\nabla_x \mathcal{L}_{CE}(x_t^{adv}, y)\right\|_1}, \quad (3)$$

and the permuted image is updated by:

$$x_{t+1}^{adv} = clip_{x,\epsilon}(x_t^{adv} + \eta \cdot sign(g_{t+1}))). \quad (4)$$

Carlini & Wagner (C&W) [2] is another powerful attack based on optimization, where an auxiliary variable ω is induced and an adversarial example constrained by l_2 norm is represented by $x' = \frac{1}{2}(\tanh\omega + 1)$. It can be optimized by:

$$\underset{\omega}{\operatorname{argmin}}\{c \cdot f(x') + \|x' - x\|_2^2\}, \quad (5)$$

where

$$f(x') = \max(\max_{i \neq y} Z(x') - Z(x')_y, -\kappa), \quad (6)$$

and here κ controls the confidence of the adversarial examples. It can also be extended to other l_p threat model by solving $c \cdot f(x + \delta) + ||\delta||_p$ in an iterative manner.

Auto Attack [3] is a combination of multiple attacks that forms a parameter-free and computationally affordable ensemble of attacks to evaluate adversarial robustness. The standard attacks includes four selected attacks: $APGD_{CE}$, targeted version of *APGD-DLR* and *FAB*, and *Square Attack*. Here we use the first two in our evaluation, because since the attack is applied in a curriculum manner, we empirically observe that after targeted *APGD-DLR*, basically

few adversarial examples are further explored by the last two attacks. So the change in the tested results of robust accuracy is quite small while the evaluation time can be significantly shortened.

References

[1] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1565–1576, 2019. 1, 2

[2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. 1, 3

[3] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning (ICML)*, 2020. 1, 3

[4] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[5] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. MMA training: Direct input space margin maximization through adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020. 1

[6] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9185–9193, 2018. 1, 3

[7] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. 2015. 1, 3

[8] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *Eighth International Conference on Learning Representations (ICLR)*, 2020. 2

[9] Byungju Kim and Junmo Kim. Adjusting decision boundary for class imbalanced learning. *IEEE Access*, pages 81674–81685, 2020. 2

[10] Saehyung Lee, Hyungyu Lee, and Sungroh Yoon. Adversarial vertex mixup: Toward better adversarially robust generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 272–281, 2020. 1

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[12] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[13] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 3

[14] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment, 2020. 1, 2

[15] Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Hang Su, and Jun Zhu. Boosting adversarial training with hypersphere embedding. 2020. 1, 2

[16] Jiawei Ren, Cunjun Yu, Shunan Sheng, Xiao Ma, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Balanced meta-softmax for long-tailed visual recognition. *Advances in Neural Information Processing Systems (NIPS)*, 2020. 1, 2

[17] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *Advances in Neural Information Processing Systems (NIPS)*, pages 3358–3369, 2019. 1, 3

[18] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 467–482. Springer, 2016. 2

[19] Kaihua Tang, Jianqiang Huang, and Hanwang Zhang. Long-tailed classification by keeping the good and removing the bad momentum causal effect. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 2

[20] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5265–5274, 2018. 2

[21] Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations (ICLR)*, 2019. 3

[22] Han-Jia Ye, Hong-You Chen, De-Chuan Zhan, and Wei-Lun Chao. Identifying and compensating for feature deviation in imbalanced deep learning. *arXiv preprin:2001.01385*, 2020. 1, 2

[23] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICLR)*, pages 7472–7482, 2019. 1