Revisiting Adversarial Training under Long-Tailed Distributions

Xinli Yue, Ningping Mou, Qian Wang, Lingchen Zhao*
Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education,
School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
{xinliyue,ningpingmou,qianwang,lczhaocs}@whu.edu.cn

Abstract

Deep neural networks are vulnerable to adversarial attacks, often leading to erroneous outputs. Adversarial training has been recognized as one of the most effective methods to counter such attacks. However, existing adversarial training techniques have predominantly been tested on balanced datasets, whereas real-world data often exhibit a long-tailed distribution, casting doubt on the efficacy of these methods in practical scenarios.

In this paper, we delve into adversarial training under long-tailed distributions. Through an analysis of the previous work "RoBal", we discover that utilizing Balanced Softmax Loss alone can achieve performance comparable to the complete RoBal approach while significantly reducing training overheads. Additionally, we reveal that, similar to uniform distributions, adversarial training under long-tailed distributions also suffers from robust overfitting. To address this, we explore data augmentation as a solution and unexpectedly discover that, unlike results obtained with balanced data, data augmentation not only effectively alleviates robust overfitting but also significantly improves robustness. We further investigate the reasons behind the improvement of robustness through data augmentation and identify that it is attributable to the increased diversity of examples. Extensive experiments further corroborate that data augmentation alone can significantly improve robustness. Finally, building on these findings, we demonstrate that compared to RoBal, the combination of BSL and data augmentation leads to a +6.66% improvement in model robustness under AutoAttack on CIFAR-10-LT. Our code is available at: https://github.com/NISPLab/AT-BSL.

1. Introduction

It is well-known that deep neural networks (DNNs) are vulnerable to adversarial attacks, where attackers can induce errors in DNNs' recognition results by adding perturbations that are imperceptible to the human eye [12, 39]. Many researchers have focused on defending against such attacks. Among the various defense methods proposed, adversarial training is recognized as one of the most effective approaches. It involves integrating adversarial examples into the training set to enhance the model's generalization capability against these examples [20, 31, 42, 45, 53, 54]. In recent years, significant progress has been made in the field of adversarial training. However, we note that almost all studies on adversarial training utilize balanced datasets like CIFAR-10, CIFAR-100 [23], and Tiny-ImageNet [24] for performance evaluation. In contrast, real-world datasets often exhibit an imbalanced, typically long-tailed distribution. Hence, the efficacy of adversarial training in practical systems should be reassessed using long-tailed datasets [14, 40].

To the best of our knowledge, RoBal [46] is the sole published work that investigates the adversarial robustness under the long-tailed distribution. However, due to its complex

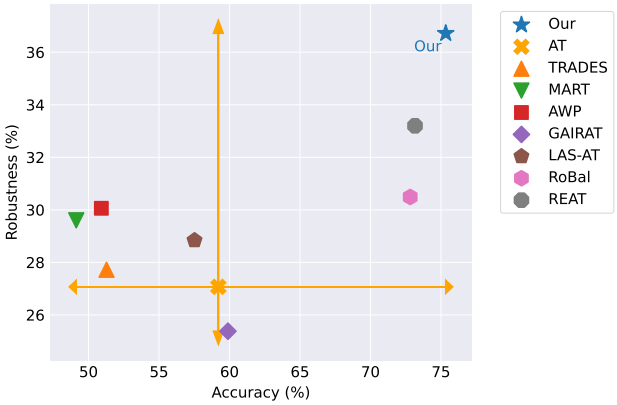


Figure 1. The clean accuracy and robustness under AutoAttack (AA) [5] of various adversarial training methods using WideResNet-34-10 [51] on CIFAR-10-LT [23]. Our method, building upon AT [31] and BSL [36], leverages data augmentation to improve robustness, achieving a +6.66% improvement over the SOTA method RoBal [46]. REAT [26] is a concurrent work with ours, yet to be published.

*Corresponding author.

design, RoBal demands extensive training time and GPU memory, which somewhat limits its practicality. Upon revisiting the design and principles of RoBal, we find that its most critical component is the Balanced Softmax Loss (BSL) [36]. We observe that combining AT [31] with BSL to form AT-BSL can match RoBal's effectiveness while significantly reducing training overhead. Following Occam's Razor principle, where entities should not be multiplied without necessity [19], we advocate using AT-BSL as a substitute for RoBal. In this paper, we base our studies on AT-BSL.

In the course of our study on the robustness of models under long-tailed distribution, we encounter another significant finding: adversarial training with long-tailed distribution data, similar to training on balanced datasets, also leads to the issue of robust overfitting [37]. Previous works on balanced datasets often employed data augmentation to mitigate this robust overfitting [4, 13, 35, 37, 45]. Hence, a straightforward approach is to attempt the use of data augmentation to alleviate the robust overfitting issue in adversarial training with long-tailed distribution. Our results align with findings on balanced datasets, indicating that data augmentation can mitigate robust overfitting. However, contrary to findings on balanced datasets where it was concluded that data augmentation alone can not improve robustness [35, 37, 45], we find that data augmentation techniques, including MixUp [52], Cutout [9], CutMix [50], AugMix [17], AutoAugment (AuA) [6], RandAugment (RA) [7] and TrivialAugment (TA) [33], can significantly improve robustness. Hence, we introduce the following query: Why does data augmentation improve robustness? We hypothesize that data augmentation augments example diversity, enabling the model to learn richer representations thereby improving its robustness. Subsequently, we validate our hypothesis through ablation studies on RA.

Our contributions are summarized as follows:

- Through ablation studies, we discover that BSL is the most critical component of RoBal, and the streamlined method AT-BSL significantly reduces training time and memory usage compared to RoBal.
- We observe that data augmentation not only mitigates robust overfitting in adversarial training under long-tailed distributions but also substantially improves robustness.
- We propose a hypothesis about the reasons for data augmentation improving robustness and validate this hypothesis through experiments.
- Comprehensive empirical evidence demonstrates that our discoveries generalize across various data augmentation strategies, model architectures, and datasets.

2. Related Works

Long-Tailed Recognition. Long-tailed distributions refer to a common imbalance in training set where a small portion of classes (head) have massive examples, while other classes (tail) have very few examples [14, 40]. Models trained under such distribution tend to exhibit a bias towards the head classes, resulting in poor performance for the tail classes. Traditional rebalancing techniques aim at addressing the long-tailed recognition problem include re-sampling [21, 38, 41, 56] and cost-sensitive learning [8, 29], which often improve the performance of tail classes at the expense of head classes. To mitigate these adverse effects, some methods handle class-specific attributes through perspectives such as margins [43] and biases [36]. Recently, more advanced techniques like class-conditional sharpness-aware minimization [57], feature clusters compression [27], and global-local mixture consistency cumulative learning [10] have been introduced, further improving the performance of long-tailed recognition. However, these works have been devoted to improving clean accuracy, and investigations into the adversarial robustness of long-tailed recognition remain scant.

Adversarial Training. The philosophy of adversarial training involves integrating adversarial examples into the training set, thereby improving the model's generalizability to such examples. Adversarial training addresses a min-max problem, with the inner maximization dedicated to generating the strongest adversarial examples and the outer minimization aimed at optimizing the model parameters. The quintessential method of adversarial training is AT [31], which can be mathematically represented as follows:

$$\begin{aligned} &\underset{\theta_m}{\operatorname{argmin}} \mathcal{L}_{\min}\left(\theta_m; x', y\right), \\ &\text{where } x' = \underset{\|x'-x\|_p \leq \epsilon}{\operatorname{argmax}} \mathcal{L}_{\max}\left(\theta_m; x', y\right). \end{aligned} \tag{1}$$

where x' is an adversarial example constrained by ℓ_p norm for clean examples x, y is the label of x, θ_m is the parameter of the model m, ϵ is the perturbation size, $\mathcal{L}_{\max}$ is the internal maximization loss, and $\mathcal{L}_{\min}$ is the external minimization loss.

Building upon the foundation of AT [31], subsequent works developed advanced adversarial training techniques such as TRADES [53], MART [42], AWP [45], GAIRAT [54], and LAS-AT [20]. However, these adversarial training methods were predominantly experimented with on balanced datasets like CIFAR-10 and CIFAR-100.

Robustness under Long-Tailed Distribution. Previous adversarial training works were concentrated mainly on balanced datasets. However, data in the real world are seldom balanced; they are more commonly characterized by long-tailed distributions [14, 40]. Therefore, a critical criterion for assessing the practical utility of adversarial training should be its performance on long-tailed distributions. To our knowledge, RoBal [46] is the only work that investigates adversarial training on long-tailed datasets. In Sec-

tion 3, we delve into the components of RoBal, improving the efficacy of long-tailed adversarial training based on our findings. Moreover, some works [30, 44, 47, 49] have already indicated that adversarial training on balanced datasets can lead to significant robustness disparities across classes. Whether this disparity is exacerbated on long-tailed datasets remains an open question for further exploration.

Data Augmentation. In standard training regimes, data augmentation has been validated as an effective tool to mitigate overfitting and improve model generalization, regardless of whether the data distribution is balanced or long-tailed [2, 10, 48, 55]. The most commonly utilized augmentation techniques for image classification tasks include random flips, rotations, and crops [15]. More sophisticated augmentation methods like MixUp [52], Cutout [9], and CutMix [50] have been shown to yield superior results in standard training contexts. Furthermore, augmentation strategies such as Augmix [17], AuA [6], RA [7], and TA [33], which employ a learned or random combination of multiple augmentations, have elevated the efficacy of data augmentation to new heights, heralding the advent of the era of automated augmentation.

3. Analysis of RoBal

3.1. Preliminaries

RoBal [46], in comparison to AT [31], incorporates four additional components: 1) cosine classifier; 2) Balanced Softmax Loss [36]; 3) class-aware margin; 4) TRADES regularization [53].

Cosine Classifier. In basic classification tasks employing a standard linear classifier, the predicted logit for class i can be represented as:

$$\begin{aligned} g(f(x))_i &= W_i^T f(x) + b_i \\ &= \|W_i\| \cdot \|f(x)\| \cos\theta_i + b_i \\ &= z_i + b_i, \end{aligned} \tag{2}$$

where $g(\cdot)$ is the liner classifer. In this formulation, the prediction depends on three factors: 1) the magnitude of the weight vector $\|W_i\|$ and the feature vector $\|f(x)\|$; 2) the angle between them, expressed as $\cos\theta_i$; and 3) the bias of the classifier b_i.

The above decomposition illustrates that simply by scaling the norm of examples in feature space, the predictions of the examples can be altered. In linear classifiers, the scale of the weight vector $\|W_i\|$ often diminishes in tail classes, thereby impacting the recognition performance for tail classes. Consequently, [46] endeavors to utilize a cosine classifier [34] to mitigate the scale effects of features and weights. And in the cosine classifier, the predicted logit for class i can be represented as:

$$\begin{aligned} h(f(x))_i &= s \cdot \left(\frac{W_i^T f(x)}{\|W_i\| \, \|f(x)\|} \right) + b_i \\ &= s \cdot \cos\theta_i + b_i, \end{aligned} \tag{3}$$

where $h(\cdot)$ is the cosine classifier, $\|\cdot\|$ denotes the ℓ_2 norm of the vector, s is the scaling factor.

Balanced Softmax Loss. An intuitive and widely adopted approach to addressing class imbalance is to assign class-specific biases during training for the cross-entropy (CE) loss. [46] employs the formulation by [32, 36], denoted as $b_i = \tau_b \log(n_i)$, where the modified cross-entropy loss, namely Balanced Softmax Loss (**BSL**), becomes:

$$\begin{aligned} &\mathcal{L}_0(h(f(x)), \mathrm{y}) = -\log\left(\frac{e^{s\cdot\cos\theta_y + b_y}}{\sum_i e^{s\cdot\cos\theta_i + b_i}} \right) \\ &= \log\left(1 + \sum_{i\neq y} e^{s\cdot(\cos\theta_i - \cos\theta_y) + \tau_b \log\left(\frac{n_i}{n_y}\right)} \right), \end{aligned} \tag{4}$$

where n_i is the number of examples in the i-th class, and τ_b is a hyperparameter controling the calculation of bias. BSL adapts to the label distribution shift between training and testing by adding specific biases to each class based on the number of examples in each class to improve long-tailed recognition performance [36].

Class-Aware Margin. However, when considering the margin representation, the margin from the true class y to class i, denoted by $\tau_b \log(n_i/n_y)$, becomes negative when $n_y > n_i$, leading to poorer discriminative representations and classifier learning for head classes. To address this, [46] introduces a class-aware margin term [34], which assigns a larger margin value to head classes as compensation:

$$m_i = \frac{\tau_m}{s} \log \frac{n_i}{n_{\min}} + m_0. \tag{5}$$

The first term increases with n_i and reaches its minimum of zero when $n_i = n_{\min}$, with τ_m as the hyperparameter controlling the trend. The second term, $m_0 > 0$, is a uniform boundary for all classes, a common strategy in networks based on cosine classifiers. Add this class-aware margin m_i to $\mathcal{L}_0$ to become $\mathcal{L}_1$:

$$\begin{aligned} &\mathcal{L}_1(h(f(x)), \mathrm{y}) \\ &= -\log\left(\frac{e^{s(\cos\theta_y - m_y) + b_y}}{e^{s(\cos\theta_y - m_y) + b_y} + \sum_{i\neq y} e^{s\cos\theta_i + b_i}} \right). \end{aligned} \tag{6}$$

TRADES Regularization. [46] incorporates a KL regularization term following TRADES [53], thereby modifying the overall loss function to:

$$\mathcal{L}_{\min} = \mathcal{L}_1\left(h\left(f\left(x'\right)\right), \mathrm{y}\right) + \beta \cdot \mathrm{KL}\left(h\left(f\left(x'\right)\right), h(f(x))\right), \tag{7}$$

where β serves as a hyperparameter to control the intensity of the TRADES regularization.

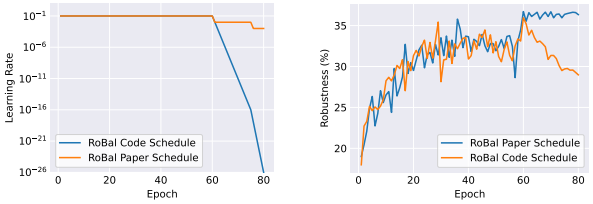
Table 1. The clean accuracy, robustness, time (average per epoch) and memory (GPU) using ResNet-18 [15] on CIFAR-10-LT following the integration of components from RoBal [46] into AT [31]. The best results are **bolded**. The second best results are underlined. Cos: Cosine Classifier; BSL: Balanced Softmax Loss [36]; CM: Class-aware Margin [46]; TRADES: TRADES Regularization [53].

Method	Components				Accuracy						Efficiency	
	Cos	BSL	CM	TRADES	Clean	FGSM	PGD	CW	LSA	AA	Time (s)	Memory (MiB)
AT [31]					54.91	32.21	28.05	28.28	28.73	26.75	21.36	**946**
AT-BSL		✓			70.21	37.44	31.91	**31.45**	**32.25**	29.48	**21.00**	**946**
AT-BSL-Cos	✓	✓			**71.99**	39.41	34.73	30.27	29.94	28.43	22.39	**946**
AT-BSL-Cos-TRADES	✓	✓		✓	69.31	39.62	34.87	30.19	30.15	28.64	38.91	1722
RoBal [46]	✓	✓	✓	✓	70.34	**40.50**	**35.93**	31.05	31.10	**29.54**	39.03	1722

3.2. Ablation Studies of RoBal

To investigate the role of each component in RoBal [46], we conduct ablation studies on it. Specifically, we incrementally add each component of RoBal to AT [31] and then evaluate the method's clean accuracy, robustness, training time per epoch, and memory usage. The results are summarized in Table 1. Note that the parameters utilized in Table 1 adhere strictly to the default settings of [46], and the details about adversarial attacks are in Section 5.1. We observe that the AT-BSL method outperforms AT [31] in terms of clean accuracy and adversarial robustness. However, upon integrating a cosine classifier with AT-BSL, while the robustness under PGD [31] significantly improves, robustness under adaptive attacks like CW [3], LSA [18], and AA [5] notably decreases. This aligns with observations in REAT [26], suggesting that the cosine classifier (scale-invariant classifier) used in RoBal may lead to gradient vanishing when generating adversarial examples with cross-entropy loss. This is attributed to the normalization of weights and features in the classification layer, which substantially reduces the gradient scale, impeding the generation of potent adversarial examples [26]. Further additions of TRADES regularization [53] and class-aware margin do not yield substantial improvements in robustness under AA, yet markedly increase training time and memory consumption. In fact, AT-BSL alone can match the complete RoBal in terms of clean accuracy and robustness under AA. Therefore, in line with Occam's Razor [19], we advocate using AT-BSL, which renders adversarial training more efficient without sacrificing significant performance. The $\mathcal{L}_{min}$ formula of AT-BSL is as follows:

$$\begin{aligned}\mathcal{L}_{\text{min}} &= \mathcal{L}_0\left(g\left(f\left(x'\right)\right), \mathrm{y}\right) \\ &= -\log\left(\frac{e^{z_y+b_y}}{\sum_i e^{z_i+b_i}}\right) \\ &= -\log\left(\frac{n_y^{\tau_b}\cdot e^{z_y}}{\sum_i n_i^{\tau_b}\cdot e^{z_i}}\right).\end{aligned} \tag{8}$$



(a) (b)

Figure 2. Learning rate scheduling analysis of RoBal [46]. (a) comparison of the learning rate schedules: 'RoBal Code Schedule' from the source code and 'RoBal Paper Schedule' as described in the publication. (b) the evolution of test robustness under PGD-20 [31] using ResNet-18 on CIFAR-10-LT across training epochs.

3.3. Robust Overfitting and Unexpected Discovery

Discrepancy in Learning Rate Scheduling: Paper Description vs. Code Implementation. RoBal [46] asserts that early stopping is not employed, and the results reported are from the final epoch, the 80th epoch. The declared learning rate schedule is an initial rate of 0.1, with decays at the 60th and 70th epochs, each by a factor of 0.1. After executing the source code of RoBal, we observe, as depicted by the blue line in Fig. 2(b), that test robustness remains essentially unchanged after the first learning rate decay (60th epoch), indicating an absence of robust overfitting. It is well-known that adversarial training on CIFAR-10 exhibits significant robust overfitting [37], and given that CIFAR-10-LT has less data than CIFAR-10, the absence of robust overfitting on CIFAR-10-LT is contradictory to the assertion that additional data can alleviate robust overfitting in [35].

Upon a meticulous examination of the official code provided by RoBal [46], we discover inconsistencies between the implemented learning rate schedule and what is claimed in the paper. The official code uses a learning rate schedule starting at 0.1, with a decay of 0.1 per epoch after the 60th epoch and 0.01 per epoch after the 75th epoch (the blue line in Fig. 2(a)). This leads to a learning rate as low as 1e-26 by the 80th epoch, potentially limiting learning after the 60th epoch and contributing to the similar performance of models at the 60th and 80th epochs as shown in Fig. 2(b).

Subsequently, we adjust the learning rate schedule to what is declared in [46] (the orange line in Fig. 2(a)) and redraw the robustness curve, represented by the orange line in Fig. 2(b). Post-adjustment, a continuous decline in test robustness following the first learning rate decay is observed, aligning with the robust overfitting phenomenon typically seen on CIFAR-10.

Therefore, adversarial training under long-tailed distributions exhibits robust overfitting, similar to balanced distributions. So, how might we resolve this problem? Several works [4, 13, 35, 37, 45] have attempted to use data augmentation to alleviate robust overfitting on balanced datasets.

Testing MixUp. [35, 37, 45] suggest that on CIFAR-10, MixUp [52] can alleviate robust overfitting. Therefore, we posit that on the long-tailed version of CIFAR-10, CIFAR-10-LT, MixUp would also mitigate robust overfitting. In Fig. 3(a), it is evident that AT-BSL-MixUp, which utilizes MixUp, significantly alleviates robust overfitting compared to AT-BSL. Furthermore, we unexpectedly discover that MixUp markedly improves robustness. This observation is inconsistent with previous findings in balanced datasets [35, 37, 45], where it was concluded that data augmentation alone does not improve robustness.

Exploring data augmentation. Following the validation of the MixUp hypothesis, our investigation expands to assess whether other augmentation techniques could alleviate robust overfitting and improve robustness. This includes augmentations like Cutout [9], CutMix [50], AugMix [17], TA [33], AuA [6], and RA [7]. Analogous to our analysis of MixUp, we report the robustness achieved by these augmentation techniques during training in Fig. 3. Firstly, our findings indicate that each augmentation technique mitigated robust overfitting, with CutMix, AuA, RA, and TA exhibiting almost negligible instances of this phenomenon. Furthermore, we observe that robustness attained by each augmentation surpasses that of the vanilla AT-BSL, further corroborating that data augmentation alone can improve robustness.

4. Why Data Augmentation Can Improve Robustness

Formulating Hypothesis. We postulate that data augmentation improves robustness by increasing example diversity, thereby allowing models to learn richer representations. Taking RA [7] as an illustrative example, for each training image, RA randomly selects a series of augmentations from a search space consisting of 14 augmentations, namely Identity, ShearX, ShearY, TranslateX, TranslateY, Rotate, Brightness, Color, Contrast, Sharpness, Posterize, Solarize, AutoContrast, and Equalize, to apply to the image. We initiate an ablation study on RA, testing the impact of each augmentation individually. Specifically, we narrow

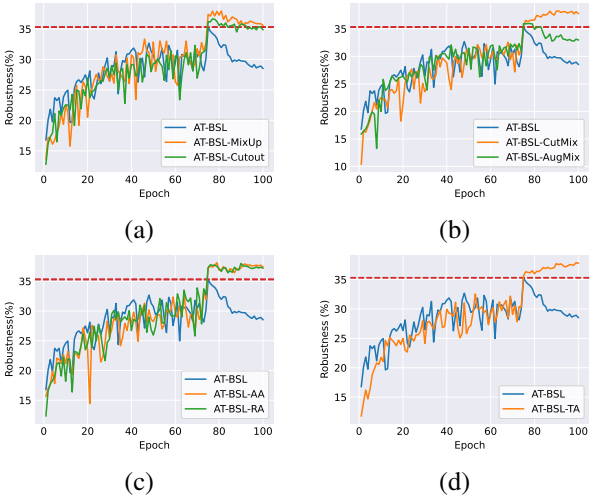
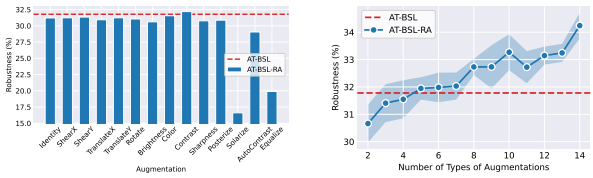


Figure 3. The evolution of test robustness under PGD-20 using ResNet-18 on CIFAR-10-LT for AT-BSL using different data augmentation strategies across training epochs. For reference, the red dashed lines in each panel represent the robustness of the best checkpoint of AT-BSL. Due to the density of the illustrations, the results have been compartmentalized into four distinct panels: (a), (b), (c), and (d).

the search space of RA to a single augmentation, meaning RA is restricted to using only this one augmentation to augment all training examples. From Fig. 4(a), it can be observed that except for Contrast, none of the augmentations alone improve robustness; in fact, augmentations such as Solarize, AutoContrast, and Equalize significantly underperform compared to AT-BSL. We surmise that this is due to the limited example diversity provided by a single augmentation, thereby resulting in no substantial improvement in robustness.

Validating Hypothesis. Subsequently, we explore the impact of the number of types of augmentations on robustness. Specifically, for each trial, we randomly selected n types of augmentations to constitute the search space of RA, with $n \in \{2, 14\}$. Each experiment is repeated five times. As shown in Fig. 4(b), we reveal that robustness progressively improves with the addition of more augmentation methods in the search space of RA. This indicates that as the number of types of augmentations in the search space increases, the variety of augmentations available to examples also grows, leading to greater example diversity. Consequently, the representations learned by the model become more comprehensive, thereby improving robustness. This validates our hypothesis.

Moreover, to further substantiate our hypothesis, we conduct an ablation study on the three types of augmentations—Solarize, AutoContrast, and Equalize—which, when used individually, impair robustness. Specifically, we eliminate these three and employ the remaining 11 augmentations as the baseline: RA-11. We then incrementally add



(a) (b)

Figure 4. The robustness under AA for AT-BSL with different augmentations using ResNet-18 on CIFAR-10-LT. (a) Change the augmentation space of RA [7] to a single augmentation, and the horizontal axis represents the name of the single augmentation. (b) The horizontal axis represents the number of types of augmentations in the search space of RA.

Table 2. The clean accuracy and robustness under AA for AT-BSL with different augmentations using ResNet-18 on CIFAR-10-LT. The best results are **bolded**. RA-11 means only using the first 11 augmentations in the search space of RA. The lines below RA-11 indicate additional augmentations based on RA-11, and the last line uses the complete search space of RA. SO: Solarize; AC: AutoContrast; EQ: Equalize.

Method	Clean	FGSM	PGD	CW	LSA	AA
RA-11	67.80	40.68	35.88	34.01	33.89	32.12
SO	67.60	41.43	37.04	34.52	34.05	32.76
AC	68.57	41.20	36.60	34.24	34.07	32.51
EQ	68.33	41.64	36.80	34.33	34.17	32.59
SO+AC	68.43	42.10	37.23	34.62	34.37	33.02
SO+EQ	68.53	41.89	37.42	35.07	34.83	33.49
AC+EQ	68.36	41.88	37.42	34.91	34.49	33.15
SO+AC+EQ	**70.86**	**43.06**	**37.94**	**36.24**	**36.04**	**34.24**

one to three of the negative augmentations, with the results outlined in Table 2. It is discovered that the more types of augmentations added, the more significant the improvement in robustness. Despite the negative impact of these three augmentations when used in isolation, their inclusion in the search space of RA still contributes to robustness improvement, further validating our hypothesis that data augmentation increases example diversity and thereby improves robustness.

5. Experiments

5.1. Settings

Datasets. Following [46], we conduct experiments on CIFAR-10-LT and CIFAR-100-LT [23]. Due to space constraints, partial results for CIFAR-100-LT are included in the appendix. In our main experiments, the imbalance ratio (IR) of CIFAR-10-LT is set to 50. Table 6 also provides results for various IRs.

Evaluation Metrics. When assessing model robustness, the l_∞ norm-bounded perturbation is $\epsilon = 8/255$. The attacks carried out include the single-step attack FGSM [12] and several iterative attacks, such as PGD [31], CW [3] and LSA [18], performed over 20 steps with a step size of $2/255$. We also employ AutoAttack (AA) [5], considered the strongest attack so far. For all methods, the evaluations are based on both the best checkpoint (selected based on robustness under PGD-20) and the final checkpoint.

Comparison Methods. We consider adversarial training methods under long-tailed distributions: RoBal [46] and REAT [26], as well as defenses under balanced distributions: AT [31], TRADES [53], MART [42], AWP [45], GAIRAT [54], and LAS-AT [20].

Training Details. We train the models using the Stochastic Gradient Descent (SGD) optimizer with an initial learning rate of 0.1, momentum of 0.9, and weight decay of 5e-4. We set the batch size to 128. We set the total number of training epochs to 100, and the learning rate is divided by 10 at the 75th and 90th epoch following [53]. During generating adversarial examples, we enforce a maximum perturbation of $8/255$ and a step size of $2/255$. The number of iterations for internal maximization is fixed at 10, denoting PGD-10, and the impact of PGD steps on robustness is investigated in Table 15. For all experiments related to AT-BSL, we adopt $\tau_b = 1$, and the results for different τ_b are provided in Fig. 7. Note that the AT-BSL presented in Tables 3 and 4 represents our own implementation, which differs in training parameters from RoBal [46]. Detailed discussions regarding these discrepancies are provided in the appendix.

5.2. Main Results

As evident from Tables 3 and 4, on CIFAR-10-LT, AT-BSL with data augmentation achieves the highest clean accuracy and adversarial robustness on both ResNet-18 and WideResNet-34-10. Note that on WideResNet-34-10, our method, AT-BSL-AuA, demonstrates a significant improvement of +6.66% robustness under AA compared to the SOTA method RoBal. Moreover, in terms of robustness at the final checkpoint, our method significantly outperforms others, demonstrating that data augmentation mitigates robust overfitting.

We present the robustness of different methods across each class in Fig. 5. It is observable that, except for a few classes, our method improves robustness in almost every class, particularly in tail classes (5 to 9 classes) where the improvements are more pronounced. Furthermore, consistent with observations on balanced datasets [30, 44, 47, 49], there is a significant disparity in class-wise robustness. Class 3 remains the least robust despite its example numbers far exceeding that of subsequent classes, which may be attributable to the intrinsic properties of class 3 [46].

5.3. Futher Analysis

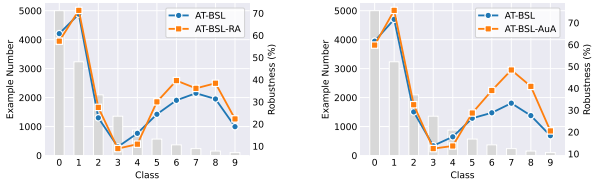
Effect of Augmentation Strategies and Parameters. We present in both Table 5 and Fig. 6 the impact of differ-

Table 3. The clean accuracy and robustness for various algorithms using ResNet-18 on CIFAR-10-LT. The best results are **bolded**.

Method	Best Checkpoint						Last Checkpoint					
	Clean	FGSM	PGD	CW	LSA	AA	Clean	FGSM	PGD	CW	LSA	AA
AT [31]	49.35	30.09	27.30	26.93	27.08	25.76	52.91	29.29	25.15	25.58	27.13	24.23
TRADES [53]	43.61	29.18	27.81	26.73	26.58	26.41	43.75	29.06	27.05	26.10	25.93	25.78
MART [42]	48.61	32.75	30.29	28.82	28.46	27.73	48.80	32.60	29.78	28.45	28.12	27.30
AWP [45]	49.29	33.78	31.20	30.53	30.36	29.53	47.75	32.77	30.83	30.01	29.68	29.12
GAIRAT [54]	50.83	30.20	27.46	21.65	21.23	20.41	50.66	28.44	25.60	19.68	19.22	18.26
LAS-AT[20]	52.81	33.35	30.32	29.57	29.15	28.53	53.50	33.14	30.09	29.13	28.84	28.30
RoBal [46]	70.34	40.50	35.93	31.05	31.10	29.54	70.00	36.18	29.00	27.67	26.98	25.63
REAT [26]	67.38	40.13	35.83	33.88	33.66	32.20	67.58	36.99	30.93	30.83	31.62	28.61
AT-BSL	68.89	40.08	35.27	33.47	33.46	31.78	67.63	35.20	28.65	28.91	31.35	26.97
AT-BSL-RA	**70.86**	**43.06**	**37.94**	**36.24**	**36.04**	**34.24**	**71.83**	**42.62**	**37.15**	**35.37**	**35.50**	**33.44**

Table 4. The clean accuracy and robustness for various algorithms using WideResNet-34-10 on CIFAR-10-LT. The best results are **bolded**.

Method	Best Checkpoint						Last Checkpoint					
	Clean	FGSM	PGD	CW	LSA	AA	Clean	FGSM	PGD	CW	LSA	AA
AT [31]	59.21	31.88	27.88	28.19	29.81	27.07	58.25	29.77	25.29	25.71	29.83	24.94
TRADES [53]	51.28	31.58	28.70	28.45	28.36	27.72	53.85	30.44	26.23	26.57	26.77	25.59
MART [42]	49.13	34.33	32.32	30.73	30.13	29.60	52.48	33.95	31.09	29.64	29.43	28.67
AWP [45]	50.91	34.28	31.85	31.23	31.01	30.06	48.65	33.21	31.07	30.33	30.14	29.40
GAIRAT [54]	59.89	33.47	30.40	26.69	26.71	25.38	56.37	29.41	27.25	23.94	23.95	23.15
LAS-AT [20]	57.52	33.66	29.86	29.60	29.44	28.84	58.19	32.98	28.89	28.75	28.58	27.90
RoBal [46]	72.82	41.34	36.42	32.48	31.95	30.49	70.85	35.95	27.74	27.59	26.76	25.71
REAT [26]	73.16	41.32	35.94	35.28	35.67	33.20	67.76	34.51	27.75	28.17	31.82	26.66
AT-BSL	73.19	41.84	35.60	34.86	35.99	32.80	65.95	33.29	27.23	27.87	31.00	26.45
AT-BSL-AuA	**75.17**	**46.18**	**40.84**	**38.82**	**39.23**	**37.15**	**77.27**	**44.73**	**38.06**	**37.14**	**39.05**	**35.11**



(a) (b)

Figure 5. The class-wise example number and robustness under AA for various algorithms on CIFAR-10-LT at the best checkpoint. (a) ResNet-18; (b) WideResNet-34-10.

ent augmentation strategies and parameters on robustness. Specifically, we conduct experiments using ResNet-18 on CIFAR-10-LT, comparing robustness at the best checkpoint. In addition, in Table 5, we use the best hyper-parameters: mixing rate $\alpha = 0.3$ for Mixup, window length 17 for Cutout, mixing rate $\alpha = 0.1$ for CutMix, and magnitude 8 for RA. As shown in Table 5, various augmentation strategies improve robustness compared to vanilla AT-BSL, with AuA and RA also achieving gains in clean accuracy. Fig. 6 indicates that for MixUp and CutMix, smaller values of α yield better robustness; for Cutout, longer window lengths

Table 5. The clean accuracy and robustness for AT-BSL with different augmentations using ResNet-18 on CIFAR-10-LT. The best results are **bolded**.

Method	Clean	FGSM	PGD	CW	LSA	AA
Vanilla	68.89	40.08	35.27	33.47	33.46	31.78
MixUp [52]	65.82	41.33	**38.05**	34.29	33.63	32.92
Cutout [9]	65.12	40.25	36.68	34.81	34.51	33.35
CutMix [50]	64.54	41.13	37.86	34.10	33.46	32.83
AugMix [17]	67.12	40.31	35.95	34.19	34.02	32.51
TA [33]	67.14	41.56	37.75	34.34	33.90	32.62
AuA [6]	**71.63**	42.69	37.78	35.60	35.47	33.69
RA [7]	70.86	**43.06**	37.94	**36.24**	**36.04**	**34.24**

generally correlate with better robustness; for RA, a moderate magnitude of transformation improves robustness, peaking at magnitude $= 8$, highlighting that excessive augmentation is not always beneficial.

Effect of Hyperparameter τ_b. To investigate the sensitivity of AT-BSL to τ_b, we evaluate the performance of AT-BSL under varying τ_b values. Specifically, we utilize

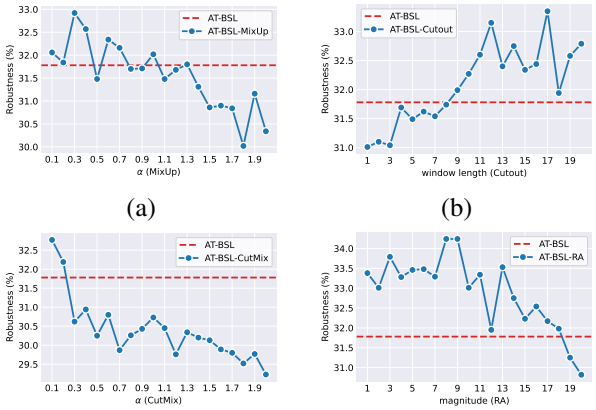


Figure 6. The robustness under AA using ResNet-18 on CIFAR-10-LT as we vary (a) the mixing rate α for MixUp, (b) the window length for Cutout, (c) the mixing rate α for CutMix, and (d) the magnitude of transformations for RA.

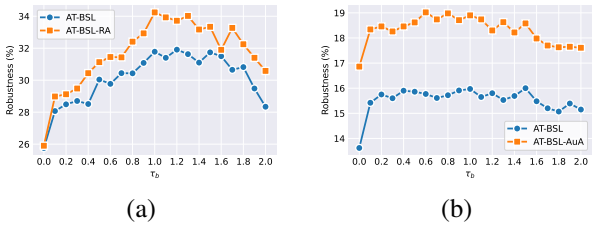


Figure 7. The robustness under AA for various algorithms with different τ_b using ResNet-18. (a): CIFAR-10-LT; (b): CIFAR-100-LT.

ResNet-18 with τ_b ranging from 0 to 20. Note that at $\tau_b = 0$, the bias $b_i = \tau_b \log(n_i)$ added by AT-BSL becomes zero, and Eq.8 reverts to the vanilla CE loss, transforming AT-BSL into vanilla AT [31]. The results, depicted in Fig. 7, reveal that on CIFAR10-LT, AT-BSL is quite sensitive to τ_b, achieving optimal robustness at $\tau_b = 1$. Conversely, on CIFAR-100-LT, AT-BSL shows less sensitivity to τ_b. Additionally, across tested datasets and τ_b values, AT-BSL with additional data augmentation consistently exhibits significantly higher robustness than vanilla AT-BSL, underscoring the substantial benefits of data augmentation in adversarial training under long-tailed distributions.

Effect of Imbalance Ratio. We further construct long-tailed datasets with varying IRs following the protocol of [8, 46] to evaluate the performance of our method. Table 6 illustrates that RA consistently improves the robustness of AT-BSL across various IR settings, further substantiating the finding that data augmentation can improve robustness.

Effect of PGD Step Size. To delve into the impact of PGD step size on robustness, we fine-tune the PGD step size from $2/255$ to $1/255$ and $0.5/255$, while also increasing the PGD steps from 10 to 20 and 40. As depicted in Table 7, it is evident that RA consistently improves the robustness of AT-BSL regardless of the PGD step size. However, we also note a decrease in robustness when compared to the baseline robustness at a PGD step size of $2/255$.

Table 6. The clean accuracy and robustness for various algorithms using ResNet-18 on CIFAR-10-LT with different imbalance ratios. Better results are **bolded**.

IR	Method	Clean	FGSM	PGD	CW	LSA	AA
10	AT-BSL	73.29	47.33	42.04	40.77	41.05	39.12
	AT-BSL-RA	**79.00**	**50.98**	**44.19**	**42.82**	**43.10**	**40.56**
20	AT-BSL	71.89	44.76	39.40	38.47	38.68	36.74
	AT-BSL-RA	**75.84**	**47.62**	**41.68**	**39.92**	**39.82**	**37.78**
50	AT-BSL	68.89	40.08	35.27	33.47	33.46	31.78
	AT-BSL-RA	**70.86**	**43.06**	**37.94**	**36.24**	**36.04**	**34.24**
100	AT-BSL	62.03	35.06	30.95	29.41	29.56	28.01
	AT-BSL-RA	**66.85**	**38.75**	**33.69**	**31.77**	**31.50**	**30.00**

Table 7. The clean accuracy and robustness for various algorithms using ResNet-18 on CIFAR-10-LT training with different PGD step sizes. Better results are **bolded**.

Size	Method	Clean	FGSM	PGD	CW	LSA	AA
0.5	AT-BSL	68.57	39.65	35.10	32.92	32.97	31.28
	AT-BSL-RA	**68.68**	**41.97**	**37.60**	**34.81**	**34.36**	**33.26**
1	AT-BSL	68.63	39.98	35.09	33.02	33.00	31.18
	AT-BSL-RA	**68.93**	**42.71**	**37.85**	**35.30**	**34.79**	**33.51**
2	AT-BSL	68.89	40.08	35.27	33.47	33.46	31.78
	AT-BSL-RA	**70.86**	**43.06**	**37.94**	**36.24**	**36.04**	**34.24**

6. Conclusion

In this paper, we first dissect the components of RoBal, identifying BSL as a critical component. We then address the issue of robust overfitting in adversarial training under long-tailed distributions and attempt to mitigate it using data augmentation. Surprisingly, we find that data augmentation not only mitigates robust overfitting but also significantly improves robustness. We hypothesize that the improved robustness is due to increased example diversity brought about by data augmentation, and we validate this hypothesis through experiments. Finally, we conduct extensive experiments with different data augmentation strategies, model architectures, and datasets, affirming the generalizability of our findings. Our findings advance adversarial training a step further towards real-world scenarios.

Acknowledgements

This work was partially supported by the NSFC under Grants U20B2049, U21B2018, and 62302344, and the Fundamental Research Funds for the Central Universities, 2042023kf0122.

References

[1] Sravanti Addepalli, Samyak Jain, et al. Efficient and effective augmentation strategy for adversarial training. In *NeurIPS*, 2022. 4

[2] Sumyeong Ahn, Jongwoo Ko, and Se-Young Yun. CUDA: Curriculum of data augmentation for long-tailed recognition. In *ICLR*, 2023. 3

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *S&P*, 2017. 4, 6, 1

[4] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019. 2, 5

[5] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 1, 4, 6

[6] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *CVPR*, 2019. 2, 3, 5, 7, 1

[7] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020. 2, 3, 5, 6, 7, 1

[8] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. 2, 8

[9] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2, 3, 5, 7, 1

[10] Fei Du, Peng Yang, Qi Jia, Fengtao Nan, Xiaoting Chen, and Yun Yang. Global and local mixture consistency cumulative learning for long-tailed visual recognitions. In *CVPR*, 2023. 2, 3

[11] Wang et al. Better diffusion models further improve adversarial training. In *ICML*, 2023. 5

[12] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 6

[13] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. 2, 5

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, 2019. 1, 2

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 4

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016. 2

[17] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *ICLR*, 2020. 2, 3, 5, 7, 1

[18] Dorjan Hitaj, Giulio Pagnotta, Iacopo Masi, and Luigi V Mancini. Evaluating the robustness of geometry-aware instance-reweighted adversarial training. *arXiv preprint arXiv:2103.01914*, 2021. 4, 6

[19] William H Jefferys and James O Berger. Ockham's razor and bayesian analysis. *American scientist*, 1992. 2, 4

[20] Xiaojun Jia, Yong Zhang, Baoyuan Wu, Ke Ma, Jue Wang, and Xiaochun Cao. Las-at: adversarial training with learnable attack strategy. In *CVPR*, 2022. 1, 2, 6, 7, 4

[21] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020. 2

[22] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 5

[23] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical report*, 2009. 1, 6

[24] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015. 1, 2

[25] Saehyung Lee, Hyungyu Lee, and Sungroh Yoon. Adversarial vertex mixup: Toward better adversarially robust generalization. In *CVPR*, 2020. 2

[26] Guanlin Li, Guowen Xu, and Tianwei Zhang. Adversarial training over long-tailed distribution. *arXiv preprint arXiv:2307.10205*, 2023. 1, 4, 6, 7, 2, 3

[27] Jian Li, Ziyao Meng, Daqian Shi, Rui Song, Xiaolei Diao, Jingwen Wang, and Hao Xu. Fcc: Feature clusters compression for long-tailed visual recognition. In *CVPR*, 2023. 2

[28] Lin Li and Michael W. Spratling. Data augmentation alone can improve adversarial training. In *ICLR*, 2023. 4, 5

[29] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 2

[30] Xinsong Ma, Zekai Wang, and Weiwei Liu. On the tradeoff between robustness and fairness. In *NeurIPS*, 2022. 3, 6

[31] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 1, 2, 3, 4, 6, 7, 8

[32] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *ICLR*, 2021. 3

[33] Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *CVPR*, 2021. 2, 3, 5, 7, 1

[34] Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Jun Zhu, and Hang Su. Boosting adversarial training with hypersphere embedding. In *NeurIPS*, 2020. 3

[35] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy A Mann. Data augmentation can improve robustness. In *NeurIPS*, 2021. 2, 4, 5, 1

[36] Jiawei Ren, Cunjun Yu, Xiao Ma, Haiyu Zhao, Shuai Yi, et al. Balanced meta-softmax for long-tailed visual recognition. In *NeurIPS*, 2020. 1, 2, 3, 4

[37] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *ICML*, 2020. 2, 4, 5

[38] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *ECCV*, 2016. 2

[39] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1

[40] 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 *CVPR*, 2018. 1, 2

[41] Tao Wang, Yu Li, Bingyi Kang, Junnan Li, Junhao Liew, Sheng Tang, Steven Hoi, and Jiashi Feng. The devil is in classification: A simple framework for long-tail instance segmentation. In *ECCV*, 2020. 2

[42] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020. 1, 2, 6, 7, 4

[43] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Learning to model the tail. In *NeurIPS*, 2017. 2

[44] Zeming Wei, Yifei Wang, Yiwen Guo, and Yisen Wang. Cfa: Class-wise calibrated fair adversarial training. In *CVPR*, 2023. 3, 6

[45] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020. 1, 2, 5, 6, 7, 4

[46] Tong Wu, Ziwei Liu, Qingqiu Huang, Yu Wang, and Dahua Lin. Adversarial robustness under long-tailed distribution. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[47] Han Xu, Xiaorui Liu, Yaxin Li, Anil Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. In *ICML*, 2021. 3, 6

[48] Zhengzhuo Xu, Zenghao Chai, and Chun Yuan. Towards calibrated model for long-tailed visual recognition from prior perspective. In *NeurIPS*, 2021. 3

[49] Xinli Yue, Ningping Mou, Qian Wang, and Lingchen Zhao. Revisiting adversarial robustness distillation from the perspective of robust fairness. In *NeurIPS*, 2023. 3, 6

[50] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 2, 3, 5, 7, 1

[51] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 1, 2, 3

[52] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 2, 3, 5, 7, 1

[53] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 1, 2, 3, 4, 6, 7

[54] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021. 1, 2, 6, 7, 4

[55] Yongshun Zhang, Xiu-Shen Wei, Boyan Zhou, and Jianxin Wu. Bag of tricks for long-tailed visual recognition with deep convolutional neural networks. In *AAAI*, 2021. 3

[56] Zizhao Zhang and Tomas Pfister. Learning fast sample reweighting without reward data. In *ICCV*, 2021. 2

[57] Zhipeng Zhou, Lanqing Li, Peilin Zhao, Pheng-Ann Heng, and Wei Gong. Class-conditional sharpness-aware minimization for deep long-tailed recognition. In *CVPR*, 2023. 2

Revisiting Adversarial Training under Long-Tailed Distributions

Supplementary Material

A. Implementation Details of Experiments

A.1. Details of Table 1

All parameters in the experiments presented in Table 1 are consistent with those used in RoBal [46]. Specifically, the initial learning rate is set at 0.1, with a decay factor of 10 applied at the 60th and 75th epochs, for a total training duration of 80 epochs. An SGD momentum optimizer is employed with a weight decay of 2×10^{-4}. The batch size is maintained at 64. For adversarial training, we adopt a maximum perturbation of $8/255$ and a step size of $2/255$, with the number of iterations for internal maximization set at 5, corresponding to PGD-5. For CIFAR-10-LT, we utilize $m_0 = 0.1$, $s = 10$, $\tau_b = 1.5$, and $\tau_m = 0.3$; for CIFAR-100-LT, the parameters are set as $m_0 = 0.3$, $s = 10$, $\tau_b = 1.5$, and $\tau_m = 0.3$. The specific hyperparameters for each experiment are detailed in Table 8.

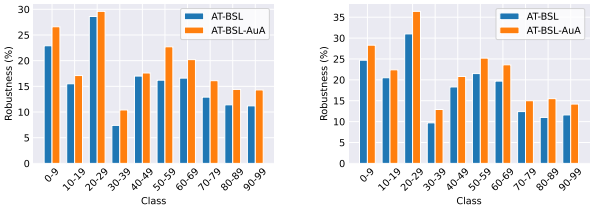
A.2. Details of Data Augmentaions

Data augmentation techniques such as MixUp [52], Cutout [9], CutMix [50], Augmix [17], AutoAugment (AuA) [6], RandAugment (RA) [7], and TrivialAugmen (TA) [33] are employed utilizing the implementations provided in torchvision 0.16.0[1]. Regarding the integration of data augmentation into the adversarial training pipeline, we follow the approach outlined in [35], whereby data augmentation precedes the generation of adversarial examples through adversarial attacks. It is observed that reversing this order, i.e., performing data augmentation after adversarial attacks, leads to the disruption of adversarial perturbations, significantly diminishing the effectiveness of the adversarial attacks.

A.3. Code References

For the defense methods compared in our paper, we utilize the official code releases, including AT [31][2], TRADES [53][3], MART [42][4], AWP [45][5], GAIRAT [54][6], LAS-AT [20][7], RoBal [46][8], and REAT [26][9]. Regarding the attacks used for evaluation, we implement them by referring to several official code repositories and the original papers, encompassing FGSM [12], PGD [31], CW [3], and AutoAttack [5][10].

[1] https://github.com/pytorch/pytorch
[2] https://github.com/MadryLab/cifar10_challenge
[3] https://github.com/yaodongyu/TRADES
[4] https://github.com/YisenWang/MART
[5] https://github.com/csdongxian/AWP
[6] https://github.com/zjfheart/Geometry-aware-Instance-reweighted-Adversarial-Training
[7] https://github.com/jiaxiaojunqaq/las-at
[8] https://github.com/wutong16/Adversarial_Long-Tail
[9] https://github.com/GuanlinLee/REAT



(a) (b)

Figure 8. The class-wise robustness under AA for various algorithms on CIFAR-100-LT at the best checkpoint. (a) ResNet-18; (b) WideResNet-34-10.

B. Extensive Experiments

B.1. More Ablation Studies of RoBal

In addition to the experiments conducted with ResNet-18 and CIFAR-10-LT as presented in Table 1, we extend our ablation studies to include WideResNet-34-10 and CIFAR-100-LT, as illustrated in Tables 9, 10, and 11. The data acquired from these additional experiments align with the conclusions drawn from Table 1, demonstrating that the AT-BSL alone achieves comparable performance in terms of clean accuracy and robustness to the complete RoBal framework. Moreover, a significant advantage is observed regarding training time and memory consumption.

B.2. Experiments on CIFAR-100-LT

Tables 12 and 13 reveal that on CIFAR-100-LT, AT-BSL with data augmentation achieves the highest clean accuracy and adversarial robustness on both ResNet-18 and WideResNet-34-10. Compared to the improvement observed on CIFAR-10-LT, the improvements on CIFAR-100-LT are less pronounced, likely due to CIFAR-100-LT's more significant number of classes and fewer examples per class, making advancements more challenging.

In Fig. 8, we illustrate the robustness of different methods across each class. Given the extremely low robustness in most classes on CIFAR-100-LT and the presence of only 50 images per class in the test set, we report the average values for every 10 classes. Notably, AuA universally improves the robustness across all class groups.

B.3. Experiments on Tiny-ImageNet-LT

To see if BSL and data augmentation are as important for higher resolution datasets as they are for low resolution

[10] https://github.com/fra31/auto-attack

Table 8. The specific hyperparameters for each experiment following the integration of components from RoBal [46] into AT [31]. Cos: Cosine Classifier; BSL: Balanced Softmax Loss [36]; CM: Class-aware Margin [46]; TRADES: TRADES Regularization [53].

Method	Components				CIFAR-10-LT				CIFAR-100-LT			
	Cos	BSL	CM	TRADES	m_0	s	τ_b	τ_m	m_0	s	τ_b	τ_m
AT [31]					0	1	0	0	0	1	0	0
AT-BSL		✓			0	1	1	0	0	1	1	0
AT-BSL-Cos	✓	✓			0.1	10	1	0	0.3	10	1	0
AT-BSL-Cos-TRADES	✓	✓		✓	0.1	10	1.5	0	0.3	10	1.5	0
RoBal [46]	✓	✓	✓	✓	0.1	10	1.5	0.3	0.3	10	1.5	0.3

Table 9. The clean accuracy, robustness, time (average per epoch), and memory (GPU) using WideResNet-34-10 [51] on CIFAR-10-LT following the integration of components from RoBal [46] into AT [31]. The best results are **bolded**. The second best results are underlined. Cos: Cosine Classifier; BSL: Balanced Softmax Loss [36]; CM: Class-aware Margin [46]; TRADES: TRADES Regularization [53].

Method	Components				Accuracy						Efficiency	
	Cos	BSL	CM	TRADES	Clean	FGSM	PGD	CW	LSA	AA	Time (s)	Memory (MiB)
AT [31]					60.86	33.22	28.79	29.24	31.27	27.66	**160.01**	**2574**
AT-BSL		✓			73.84	39.13	32.02	32.29	**34.98**	30.21	162.01	**2574**
AT-BSL-Cos	✓	✓			**74.69**	40.86	34.77	31.14	30.50	29.22	163.71	**2574**
AT-BSL-Cos-TRADES	✓	✓		✓	73.34	41.28	**36.49**	31.79	31.55	30.05	302.62	6932
RoBal [46]	✓	✓	✓	✓	72.82	**41.34**	36.42	**32.48**	31.95	**30.49**	309.09	6932

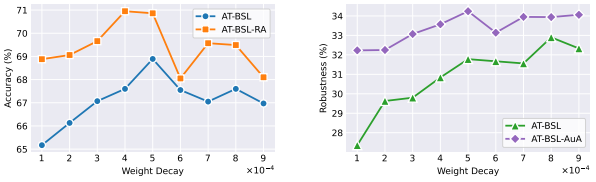
datasets (such as CIFAR-10-LT and CIFAR-100-LT), we conduct experiments on Tiny-ImageNet [24]. Firstly, Tiny-ImageNet is a dataset consisting of 200 classes, with images sized 64*64 pixels, making it four times the resolution of CIFAR-10/100. We derive Tiny-ImageNet-LT using an IR of 0.1 from Tiny-ImageNet. Following [20, 25], we employ the PreActResNet-18 model [16]. Apart from the model, the experimental setup for Tiny-ImageNet-LT remains largely similar to that of CIFAR-10-LT. As observed from the Table 14, both BSL and data augmentation prove to be crucial for Tiny-ImageNet-LT.

B.4. Different PGD Steps

To investigate the impact of PGD steps on robustness, we assess the robustness achieved using different PGD steps following [46]. Table 15 indicates that RA consistently improves the robustness of AT-BSL regardless of PGD steps, and the clean accuracy also experiences improvement. Moreover, there is a trade-off between clean accuracy and robustness: as the PGD step increases, clean accuracy decreases while robustness improves. The optimal trade-off is attained at PGD-10. Hence, we employ PGD-10 in our experiments involving AT-BSL.

B.5. Different Weight Decay

During our replication of the experiments of REAT [26], we observe a discrepancy in the weight decay parameters used: REAT employed a weight decay of 5×10^{-4}, contrasting



(a) (b)

Figure 9. The clean accuracy and robustness under AA for various algorithms with different weight decay using ResNet-18 on CIFAR-10-LT at the best checkpoint.

with 2×10^{-4} used by RoBal [46]. This leads us to conduct experiments using varying values of weight decay. The results, depicted in Fig. 9, indicate that a weight decay of 5×10^{-4} offers a significant improvement over 2×10^{-4} in terms of both accuracy and robustness. However, further increasing the weight decay beyond 5×10^{-4} results in a noticeable decline in accuracy. Therefore, we employ a weight decay of 5×10^{-4} in our experiments.

B.6. Different Batch Sizes

While replicating the experiments of REAT [26], we note an inconsistency in the batch size settings: REAT utilized a batch size of 128, whereas RoBal utilized 64. To address this, we conduct experiments with different batch sizes, and the results are presented in Table 16. The findings indicate that the performance with batch sizes 64 and 128 are comparable, and both outperform larger batch sizes; however,

Table 10. The clean accuracy, robustness, time (average per epoch) and memory (GPU) using ResNet-18 [15] on CIFAR-100-LT following the integration of components from RoBal [46] into AT [31]. The best results are **bolded**. The second best results are underlined. Cos: Cosine Classifier; BSL: Balanced Softmax Loss [36]; CM: Class-aware Margin [46]; TRADES: TRADES Regularization [53].

Method	Components				Accuracy						Efficiency	
	Cos	BSL	CM	TRADES	Clean	FGSM	PGD	CW	LSA	AA	Time (s)	Memory (MiB)
AT [31]					44.32	18.81	15.11	15.36	17.85	13.91	43.25	**946**
AT-BSL		✓			45.78	21.58	**18.96**	17.78	18.48	16.35	**41.99**	**946**
AT-BSL-Cos	✓	✓			41.83	17.95	14.69	14.22	14.87	13.14	43.86	**946**
AT-BSL-Cos-TRADES	✓	✓		✓	37.50	16.92	14.05	13.98	14.52	12.87	72.34	1724
RoBal [46]	✓	✓	✓	✓	**45.93**	**21.35**	17.40	**17.80**	**19.14**	**16.42**	72.93	1724

Table 11. The clean accuracy, robustness, time (average per epoch), and memory (GPU) using WideResNet-34-10 [51] on CIFAR-100-LT following the integration of components from RoBal [46] into AT [31]. The best results are **bolded**. The second best results are underlined. Cos: Cosine Classifier; BSL: Balanced Softmax Loss [36]; CM: Class-aware Margin [46]; TRADES: TRADES Regularization [53].

Method	Components				Accuracy						Efficiency	
	Cos	BSL	CM	TRADES	Clean	FGSM	PGD	CW	LSA	AA	Time (s)	Memory (MiB)
AT [31]					48.87	21.14	17.20	17.61	21.23	16.27	**319.33**	**2574**
AT-BSL		✓			49.68	**23.08**	**19.81**	**19.47**	21.19	17.84	323.66	**2574**
AT-BSL-Cos	✓	✓			48.29	20.25	16.34	16.43	17.90	15.09	327.17	**2574**
AT-BSL-Cos-TRADES	✓	✓		✓	44.37	18.94	15.48	15.70	17.02	14.43	603.99	6936
RoBal [46]	✓	✓	✓	✓	**50.08**	23.04	18.84	19.30	**21.87**	**17.90**	617.73	6936

128 is more commonly used and helps speed up training. Consequently, we employ a batch size of 128 in our experiments.

B.7. Different Training Epochs

As indicated in Table 17, without data augmentation, the results between 80 and 100 training epochs show little difference. However, with data augmentation, we observe that a higher number of training epochs leads to increased robustness. This improvement is likely attributable to the augmented diversity of examples, necessitating a more extended learning period for the model.

B.8. Hyperparameter Tuning of RoBal

Through hyperparameter tuning similar to those done for AT-BSL using ResNet-18 on CIFAR-10-LT, we find that RoBal achieve the best results with PGD-10, weight decay of 2×10^{-4}, batch size of 64, epochs of 60, and $\tau_b = 1.5$. The robustness under AA reach 31.61%, which is close to the performance of AT-BSL.

B.9. Retraining RoBal and REAT

Compared to RoBal [46], our primary experiments employ different experimental settings, including previously discussed variables like PGD steps, weight decay, batch size, and training epochs. To facilitate a fairer comparison, we adapt these settings in our main experiments: changing PGD-5 to PGD-10, weight decay from 2×10^{-4} to 5×10^{-4}, batch size from 64 to 128, and increasing training epochs from 80 to 100, and then we retrain RoBal under these settings, referred to as RoBal (retraining). Compared with REAT [26], the only discrepancy is in the training epochs. Therefore, we adjusted REAT's training epochs to 100 and conduct a retraining called REAT (retraining). The results are presented in Table 18. The retrained RoBal is observed to achieve improved robustness, albeit at a slight cost to accuracy. Conversely, the retrained REAT displays even lower robustness than its initial version. Through this comparison, we note that the robustness achieved by the retrained RoBal and REAT is similar to that of the vanilla AT-BSL.

B.10. Other Data Augmentations

Data Augmentations Designed for Long-Tailed Recognition. CUDA [2][11] initially explored the relationship between the degree of augmentation and class performance, discovering that an appropriate level of augmentation needs to be allocated for each class to mitigate class imbalance issues. Inspired by this finding, [2] introduces a simple yet efficient novel curriculum to identify the appropriate data augmentation strength for each class, called CUDA: CUrriculum of Data Augmentation for long-tailed recognition. To assess CUDA's performance in adversarial training under long-tailed distributions, we augment AT-BSL with CUDA, referred to as AT-BSL-CUDA, and compared

[11] https://github.com/sumyeongahn/cuda_ltr

Table 12. The clean accuracy and robustness for various algorithms using ResNet-18 on CIFAR-100-LT. The best results are **bolded**.

Method	Best Checkpoint						Last Checkpoint					
	Clean	FGSM	PGD	CW	LSA	AA	Clean	FGSM	PGD	CW	LSA	AA
AT [31]	41.20	17.42	14.59	14.51	16.49	13.62	41.44	17.21	13.89	14.17	16.40	13.10
TRADES [53]	38.12	19.60	17.89	15.96	15.91	15.59	38.71	19.43	17.27	15.83	15.87	15.28
MART [42]	38.46	23.04	21.36	18.59	18.36	17.51	39.58	22.38	20.51	18.40	18.42	17.27
AWP [45]	41.53	23.47	21.79	19.68	19.73	18.61	43.57	22.91	20.72	19.11	19.30	17.82
GAIRAT [54]	38.99	19.73	18.05	16.59	16.80	15.61	39.70	14.66	11.87	11.57	12.28	10.48
LAS-AT [20]	44.33	22.02	19.59	17.18	17.11	16.15	44.70	22.11	19.23	16.93	17.03	15.77
RoBal [46]	45.93	21.35	17.40	17.80	19.14	16.42	45.78	19.97	15.37	15.75	18.67	14.51
REAT [26]	46.28	21.55	18.85	18.07	18.95	16.54	45.99	19.62	16.29	16.04	18.22	14.79
AT-BSL	45.59	21.14	18.05	17.34	18.14	15.97	45.35	18.96	15.52	15.59	17.78	14.41
AT-BSL-AuA	**48.39**	**25.81**	**22.96**	**20.73**	**21.30**	**18.90**	**50.66**	**25.89**	**22.43**	**20.62**	**21.43**	**18.79**

Table 13. The clean accuracy and robustness for various algorithms using WideResNet-34-10 on CIFAR-100-LT. The best results are **bolded**.

Method	Best Checkpoint						Last Checkpoint					
	Clean	FGSM	PGD	CW	LSA	AA	Clean	FGSM	PGD	CW	LSA	AA
AT [31]	45.18	19.25	16.36	16.43	19.00	15.60	44.86	19.01	15.65	15.89	19.12	15.08
TRADES [53]	41.71	21.91	19.85	18.46	18.39	17.91	43.22	20.28	17.46	17.34	17.56	16.69
MART [42]	41.32	25.01	23.27	20.89	20.77	19.98	43.67	22.84	19.88	18.80	19.45	17.77
AWP [45]	45.66	25.89	23.88	21.87	22.10	20.56	48.18	24.75	21.81	20.30	21.19	18.67
GAIRAT [54]	36.41	18.87	17.31	16.07	16.13	14.77	45.11	19.49	16.31	15.85	16.71	14.75
LAS-AT[20]	45.86	23.30	20.02	18.67	18.79	17.35	46.54	22.84	19.65	18.18	18.38	17.01
RoBal [46]	50.08	23.04	18.84	19.30	21.87	17.90	46.34	19.99	15.17	15.87	20.06	14.77
REAT [26]	50.29	23.99	20.82	20.25	21.93	18.65	49.22	20.89	16.57	17.08	20.89	15.49
AT-BSL	50.04	23.37	19.66	19.60	21.66	18.04	48.56	20.88	16.83	17.09	20.13	15.76
AT-BSL-AuA	**53.08**	**28.55**	**25.40**	**23.39**	**24.48**	**21.43**	**55.55**	**26.74**	**22.18**	**21.88**	**24.28**	**19.68**

Table 14. The robustness for various algorithms with different training epochs using PreActResNet-18 on Tiny-ImageNet-LT at the best checkpoint. Better results are **bolded**.

Method	Clean	FGSM	PGD	CW	LSA	AA
AT	36.30	16.58	14.52	12.65	13.16	11.37
RoBal	36.27	13.66	10.98	10.18	9.84	8.98
AT-BSL	38.83	17.47	15.34	13.35	14.08	11.83
AT-BSL-RA	**39.00**	**18.82**	**16.94**	**14.26**	**14.60**	**12.73**

it with the vanilla AT-BSL, as shown in the Table 19. The results suggest that CUDA's performance in adversarial training under long-tailed distributions appears less effective than RA.

Data Augmentations Designed for Adversarial Training. DAJAT [1][12] proposes a data augmentation technique designed explicitly for adversarial training. [1] initially conceptualizes data augmentation as a domain generalization problem during the training process. Subsequently, they introduce Diverse Augmentation-based Joint Adversarial Training (DAJAT), effectively integrating data augmentation into adversarial training. Since DAJAT's experiments are based on TRADES [53], it cannot be directly applied to augment AT-BSL. We conduct comparative analyses between vanilla TRADES and DAJAT. The comparison in Table 19 reveals that DAJAT still contributes to improved robustness in long-tailed adversarial training, showing comparable effectiveness to TRADES-RA.

IDBH [28][13] is another data augmentation technique that is specifically formulated for adversarial training. [28] discovers that the diversity and hardness of data augmentation play a crucial role in combating adversarial overfitting. Overall, diversity enhances both accuracy and robustness, while hardness can improve robustness to a certain extent, but at the expense of accuracy and beyond a certain thresh-

[12] https://github.com/val-iisc/dajat

[13] https://github.com/treelli/da-alone-improves-at

Table 15. The clean accuracy and robustness for various algorithms using ResNet-18 on CIFAR-10-LT training with different PGD steps. Better results are **bolded**.

Steps	Method	Clean	FGSM	PGD	CW	LSA	AA
1	AT-BSL	77.15	23.15	12.05	13.06	24.80	11.27
	AT-BSL-RA	**82.16**	**28.28**	**14.25**	**15.34**	**26.30**	**13.21**
3	AT-BSL	72.37	36.61	28.95	28.79	30.23	26.64
	AT-BSL-RA	**74.20**	**40.39**	**32.63**	**32.25**	**33.31**	**29.79**
5	AT-BSL	68.62	39.11	33.67	32.47	32.62	30.49
	AT-BSL-RA	**69.39**	**41.86**	**36.81**	**34.33**	**33.89**	**32.62**
7	AT-BSL	68.28	39.55	34.62	32.94	32.68	31.16
	AT-BSL-RA	**68.79**	**42.45**	**37.78**	**35.31**	**34.98**	**33.57**
10	AT-BSL	68.89	40.08	35.27	33.47	33.46	31.78
	AT-BSL-RA	**70.86**	**43.06**	**37.94**	**36.24**	**36.04**	**34.24**
11	AT-BSL	67.89	39.78	35.21	33.15	33.20	31.57
	AT-BSL-RA	**68.46**	**42.10**	**37.63**	**34.58**	**34.26**	**33.12**
13	AT-BSL	**69.07**	39.82	35.19	33.12	32.91	31.44
	AT-BSL-RA	68.90	**42.18**	**37.89**	**34.93**	**34.58**	**33.35**

Table 16. The robustness for various algorithms with different batch sizes using ResNet-18 on CIFAR-10-LT at the best checkpoint. Better results are **bolded**. BS: Batch Size.

BS	Method	Clean	FGSM	PGD	CW	LSA	AA
64	AT-BSL	**67.82**	41.42	36.57	34.41	34.22	32.67
	AT-BSL-RA	66.70	**41.85**	**37.87**	**35.34**	**34.83**	**33.78**
128	AT-BSL	68.89	40.08	35.27	33.47	33.46	31.78
	AT-BSL-RA	**70.86**	**43.06**	**37.94**	**36.24**	**36.04**	**34.24**
256	AT-BSL	66.72	37.66	33.08	31.55	31.42	29.98
	AT-BSL-RA	**67.93**	**41.05**	**36.60**	**33.78**	**33.43**	**31.97**
512	AT-BSL	60.01	35.45	32.27	29.44	29.02	28.15
	AT-BSL-RA	**63.25**	**37.66**	**34.38**	**31.14**	**30.59**	**29.53**

Table 17. The robustness for various algorithms with different training epochs using ResNet-18 on CIFAR-10-LT at the best checkpoint. Better results are **bolded**.

Method	Clean	FGSM	PGD	CW	LSA	AA
AT-BSL-80	66.68	**40.18**	**36.11**	**33.87**	**33.64**	**31.95**
AT-BSL	**68.89**	40.08	35.27	33.47	33.46	31.78
AT-BSL-RA-80	69.39	41.93	37.20	34.82	34.36	32.92
AT-BSL-RA	**70.86**	**43.06**	**37.94**	**36.24**	**36.04**	**34.24**

old, it diminishes both. [28] introduces a novel cropping transformation method called Cropshift to mitigate robust overfitting. Building on Cropshift, [28] proposes a new augmentation scheme called Improved Diversity and Balanced Hardness (IDBH). We utilize IDBH to augment AT-BSL, referred to as AT-BSL-IDBH. Upon comparison in Table 19, it is found that IDBH's effectiveness is less pronounced than RA on long-tailed datasets.

Table 18. The robustness for various algorithms using ResNet-18 on CIFAR-10-LT at the best checkpoint. The best results are **bolded**.

Method	Clean	FGSM	PGD	CW	LSA	AA
RoBal	70.34	40.50	35.93	31.05	31.10	29.54
RoBal (retraining)	67.46	41.61	**38.04**	32.75	33.08	31.26
REAT	67.38	40.13	35.83	33.88	33.66	32.20
REAT (retraining)	67.38	39.51	35.15	33.53	33.31	31.77
AT-BSL	68.89	40.08	35.27	33.47	33.46	31.78
AT-BSL-RA	**70.86**	**43.06**	37.94	**36.24**	**36.04**	**34.24**

Table 19. The robustness for various algorithms with different data augmentations using ResNet-18 on CIFAR-10-LT at the best checkpoint. The best results are **bolded**.

Method	Clean	FGSM	PGD	CW	LSA	AA
TRADES	43.61	29.18	**27.81**	**26.73**	**26.58**	**26.41**
DAJAT	42.04	**29.34**	27.70	26.47	26.36	26.27
TRADES-RA	**44.45**	29.18	27.61	26.51	26.47	26.27
AT-BSL	68.89	40.08	35.27	33.47	33.46	31.78
AT-BSL-CUDA	68.05	40.06	36.48	33.07	32.75	31.49
AT-BSL-IDBH	70.80	39.54	33.30	32.56	33.01	31.24
AT-BSL-RA	**70.86**	**43.06**	**37.94**	**36.24**	**36.04**	**34.24**

Table 20. The robustness for various algorithms with different training epochs using ResNet-18 on CIFAR-10-LT at the best checkpoint. Better results are **bolded**.

Method	Clean	FGSM	PGD	CW	LSA	AA
AT-BSL	68.89	40.08	35.27	33.47	33.46	31.78
AT-BSL-RA	70.86	43.06	37.94	36.24	36.04	34.24
AT-BSL-DM	**72.61**	**47.09**	**42.01**	**41.56**	**41.89**	**39.48**

B.11. Using Data Generated by Diffusion Models

To investigate the potential of leveraging data generated by diffusion models to improve the robustness of AT-BSL, we train a diffusion model, DDPM++, for CIFAR-10-LT, selecting the version with the best Fréchet Inception Distance (FID) of 6.92 after 18 sampling steps following [11, 22]. For the generation of 1 million data points, we produce 100,000 images per class, culminating in a total of 1 million images. Following [11], we set the proportion of unsupervised data to 0.7 and train a ResNet-18 using AT-BSL, which we refer to as AT-BSL-DM. The results presented in Table 20 clearly demonstrate the significant improvement in robustness afforded by incorporating data generated by diffusion models.

B.12. Different Adversarial Training Methods

To further validate the hypothesis that data augmentation alone improves robustness under long-tailed distributions, we conduct experiments across various adversarial training methods, employing AuA or RA. As evidenced in Table 21, with few exceptions, data augmentation is beneficial for robustness. This effect is particularly common on CIFAR-100-LT, likely due to the reduced number of training examples per class in this dataset, leading to a more substantial reliance on data augmentation techniques.

B.13. Standard Deviation

We repeat AT-BSL and AT-BSL-RA five times using ResNet-18 on CIFAR-10-LT. Their mean and standard deviation of robustness under AA are 31.65 ± 0.45 and 34.12 ± 0.51, respectively. The relatively small variance indicates the stability of our training process.

B.14. Computational Cost Comparison

In this section, we compare the computational costs of AT-BSL and AT-BSL-RA/AuA regarding average training time per epoch and GPU memory usage. The detailed results are summarized in Table 22. The comparison indicates that the introduction of data augmentation incurs a negligible increase in time cost without imposing additional memory overhead.

C. Comparison with Concurrent Works

Concurrently and independently from our work, REAT [26] has also explored adversarial training under long-tailed distributions. [26] identifies that compared to conventional adversarial training on balanced datasets, this process tends to produce imbalanced adversarial examples and feature embedding spaces, resulting in reduced robustness on tail data. To address this issue, [26] introduces a novel adversarial training framework: Re-balancing Adversarial Training (REAT). This framework comprises two key components: (1) a new training strategy inspired by the concept of effective numbers, guiding the model to generate more balanced and informative adversarial examples, and (2) a meticulously designed penalty function aimed at enforcing a satisfactory feature space. Notably, the experimental settings utilized in our paper are fundamentally consistent with those employed in REAT. Moreover, as shown in Tables 3, 4, 12, and 13, the robustness achieved by our implemented vanilla AT-BSL is comparable to that of REAT.

Table 21. The robustness for various algorithms with/without data augmentations at the best checkpoint. On the combination of ResNet-18 and CIFAR-10-LT, RA is employed, whereas, for other model and dataset combinations, AuA is utilized. Better results are **bolded**.

Method	CIFAR-10-LT						CIFAR-100-LT					
	ResNet-18			WideResNet-34-10			ResNet-18			WideResNet-34-10		
	Clean	PGD	AA	Clean	PGD	AA	Clean	PGD	AA	Clean	PGD	AA
AT [31]	**49.35**	27.30	25.76	59.21	27.88	27.07	41.20	14.59	13.62	45.18	16.36	15.60
AT-RA/AuA	44.31	**27.81**	**25.90**	**62.98**	**33.40**	**31.64**	**45.17**	**19.78**	**17.22**	**50.00**	**21.87**	**19.44**
TRADES [53]	43.61	**27.81**	**26.41**	51.28	28.70	27.72	38.12	17.89	15.59	41.71	19.85	17.91
TRADES-RA/AuA	**44.45**	27.61	26.27	**55.89**	**31.53**	**29.77**	**42.14**	**19.69**	**16.12**	**46.23**	**22.78**	**19.52**
MART [42]	**48.61**	**30.29**	**27.73**	**49.13**	**32.32**	**29.60**	**38.46**	21.36	17.51	41.32	23.27	19.98
MART-RA/AuA	43.76	29.86	26.77	48.07	31.93	28.31	38.01	**22.64**	**18.68**	**43.43**	**25.41**	**21.26**
AWP [45]	**49.29**	**31.20**	**29.53**	**50.91**	**31.85**	**30.06**	**41.53**	21.79	18.61	**45.66**	23.88	20.56
AWP-RA/AuA	45.28	30.56	28.73	44.06	29.91	27.81	41.07	**23.02**	**19.37**	45.27	**25.76**	**21.60**
GAIRAT [54]	**50.83**	**27.46**	**20.41**	59.89	30.40	25.38	38.99	18.05	**15.61**	36.41	17.31	14.77
GAIRAT-RA/AuA	43.56	27.34	17.82	**66.43**	**37.96**	**25.53**	**41.94**	**19.18**	14.82	**49.75**	**22.19**	**18.24**
LAS-AT [20]	**52.81**	30.32	28.53	57.52	29.86	28.84	44.33	19.59	16.15	45.86	20.02	17.35
LAS-AT-RA/AuA	51.20	**31.20**	**29.18**	**59.14**	**34.51**	**32.54**	**45.18**	**22.78**	**18.61**	**49.73**	**24.09**	**20.79**
RoBal [46]	**70.34**	35.93	29.54	**72.82**	36.42	30.49	45.93	17.40	16.42	50.08	18.84	17.90
RoBal-RA/AuA	68.66	**37.50**	**30.06**	72.57	**40.54**	**31.87**	**47.75**	**19.93**	**18.04**	**54.12**	**21.41**	**19.66**
REAT [26]	**67.38**	35.83	**32.20**	**73.16**	35.94	33.20	46.28	18.85	16.54	50.29	20.82	18.65
REAT-RA/AuA	66.64	**36.97**	31.84	72.05	**40.05**	**35.74**	**47.65**	**22.86**	**18.48**	**50.10**	**25.07**	**20.81**
AT-BSL	68.89	35.27	31.78	73.19	35.60	32.80	45.59	18.05	15.97	50.04	19.66	18.04
AT-BSL-RA/AuA	**70.86**	**37.94**	**34.24**	**75.17**	**40.84**	**37.15**	**48.39**	**22.96**	**18.90**	**53.08**	**25.40**	**21.43**

Table 22. The time and memory for various algorithms. On the combination of ResNet-18 and CIFAR-10-LT, RA is employed, whereas, for other model and dataset combinations, AuA is utilized. All experiments are run on NVIDIA RTX 3090.

Dataset	Method	ResNet-18		WideResNet-34-10	
		Time (s)	Memory (MiB)	Time (s)	Memory (MiB)
CIFAR-10-LT	AT-BSL	22.37	1345	200.70	4293
	AT-BSL-RA/AuA	22.43	1345	201.42	4293
CIFAR-100-LT	AT-BSL	30.94	1347	277.82	4293
	AT-BSL-RA/AuA	31.25	1347	279.66	4293