Appendix for paper *Advancing Example Exploitation Can Alleviate Critical Challenges in Adversarial Training*

A. Illustration of A-C/R-C examples

We illustrate some A-C/R-C examples in Figure 7 and Figure 8. Considering the visual representation, it can be found that the features in R-C examples are more salient than in A-C examples.

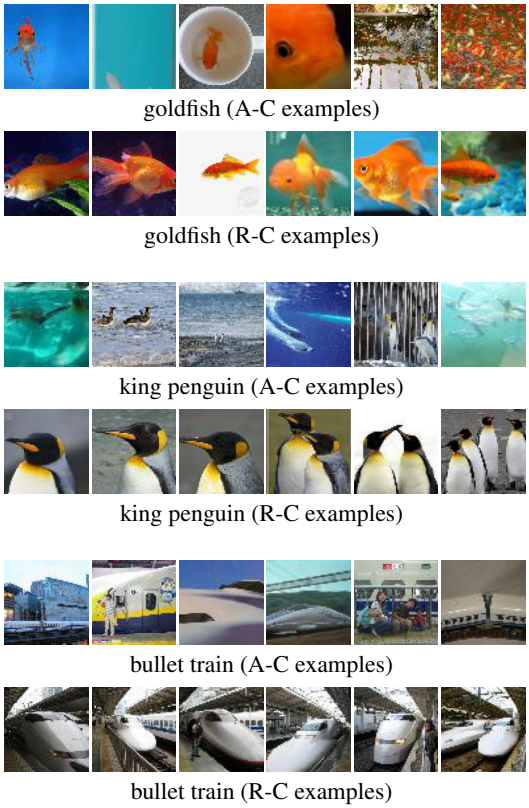


Figure 7: Illustration of A-C/R-C examples in TinyImageNet dataset.

B. Other example-exploitation AT methods

B.1. Treatments of examples

In Table 3, we give the loss functions of some AT methods. Most of them are implemented in two versions: PGDAT-based or TRADES-based. We tend to select the TRADES-based implementations for disentangling the accuracy and robustness. In the following, we analyze some methods that appeared in Table 1 for their treatments.

Table 3: The loss functions of different AT methods.

PGDAT: $\mathrm{CE}\left(\mathbf{p}\left(\boldsymbol{x}'\right), y\right)$
TRADES: $\mathrm{CE}(\mathbf{p}(\boldsymbol{x}), y) + \lambda \cdot \mathrm{KL}\left(\mathbf{p}(\boldsymbol{x}) \| \mathbf{p}\left(\boldsymbol{x}'\right)\right)$
SAT: $w_i^{\mathrm{S}} \cdot \mathrm{CE}\left(\mathbf{p}\left(\boldsymbol{x}_i\right), \boldsymbol{t}_i^{\mathrm{S}}\right) + \lambda \cdot \mathrm{KL}\left(\mathbf{p}\left(\boldsymbol{x}_i\right) \| \mathbf{p}\left(\boldsymbol{x}_i'\right)\right)$
MART: $\mathrm{BCE}\left(\mathbf{p}\left(\boldsymbol{x}'\right), y\right) + \lambda \cdot \mathrm{KL}\left(\mathbf{p}(\boldsymbol{x}) \| \mathbf{p}\left(\boldsymbol{x}'\right)\right) \cdot \left(1 - \mathbf{p}_y(\boldsymbol{x})\right)$
FAT: $\mathrm{CE}\left(\mathbf{p}\left(\boldsymbol{x}_i\right), \boldsymbol{y}_i\right) + \lambda \cdot \mathrm{KL}\left(\mathbf{p}\left(\boldsymbol{x}_i\right) \| \mathbf{p}\left(\boldsymbol{x}_i'^{\mathrm{F}}\right)\right)$
GAIRAT: $w_i^{\mathrm{G}} \cdot \mathrm{CE}\left(\mathbf{p}\left(\boldsymbol{x}_i'\right), \boldsymbol{y}_i\right)$
TEAT: $\mathrm{CE}\left(\mathbf{p}\left(\boldsymbol{x}_i\right), \boldsymbol{y}_i\right) + \lambda \cdot \mathrm{KL}\left(\mathbf{p}\left(\boldsymbol{x}_i\right) \| \mathbf{p}\left(\boldsymbol{x}_i'^{\mathrm{T}}\right)\right) + w \cdot \left\|\boldsymbol{t}_i^{\mathrm{T}} - \mathbf{p}\left(\boldsymbol{x}_i\right)\right\|_2^2$
RC-TRADES: $\mathrm{CE}\left(\mathbf{p}\left(\boldsymbol{x}_i\right), \boldsymbol{y}_i\right) + \lambda_i \cdot \mathrm{KL}\left(\mathbf{p}\left(\boldsymbol{x}_i\right) \| \mathbf{p}\left(\boldsymbol{x}_i'\right)\right)$
RC-TEAT: $\mathrm{CE}\left(\mathbf{p}\left(\boldsymbol{x}_i\right), \boldsymbol{y}_i\right) + \lambda_i \cdot \mathrm{KL}\left(\mathbf{p}\left(\boldsymbol{x}_i\right) \| \mathbf{p}\left(\boldsymbol{x}_i'\right)\right) + w \cdot \left\|\boldsymbol{t}_i^{\mathrm{T}} - \mathbf{p}\left(\boldsymbol{x}_i\right)\right\|_2^2$

SAT: Compared with TRADES, SAT keeps the KL term unchanged but modifies the CE term for each example. Besides replacing the original target of $\boldsymbol{x}_i$ with $\boldsymbol{t}_i^{\mathrm{S}}$, SAT also reweights the CE term with factor w_i^{S} (w_i^{S} is small for $\boldsymbol{x}_i$ which is easily misclassified). These two schemes limit the accuracy learning of the model for A-C examples.

MART: MART uses boosted cross-entropy (BCE) instead of the commonly used CE loss: $\mathrm{BCE}(\mathbf{p}(x'), y) = -\log(\mathbf{p}_y(\boldsymbol{x}')) - \log(1 - \max_{k \neq y} \mathbf{p}_k(\boldsymbol{x}'))$. Through reweighting the KL term with the adaptive value $1 - \mathbf{p}_y(\boldsymbol{x})$ (large for easily misclassified examples), MART enhances the robustness learning of the model for A-C examples.

FAT: The only difference between FAT and TRADES is that FAT generates different adversarial examples ($\boldsymbol{x}_i'^{\mathrm{F}}$) for training by using fewer attack iteration steps. The attacker needs more steps for R-C examples than A-C examples to generate strong enough adversarial perturbations that lead to mis-

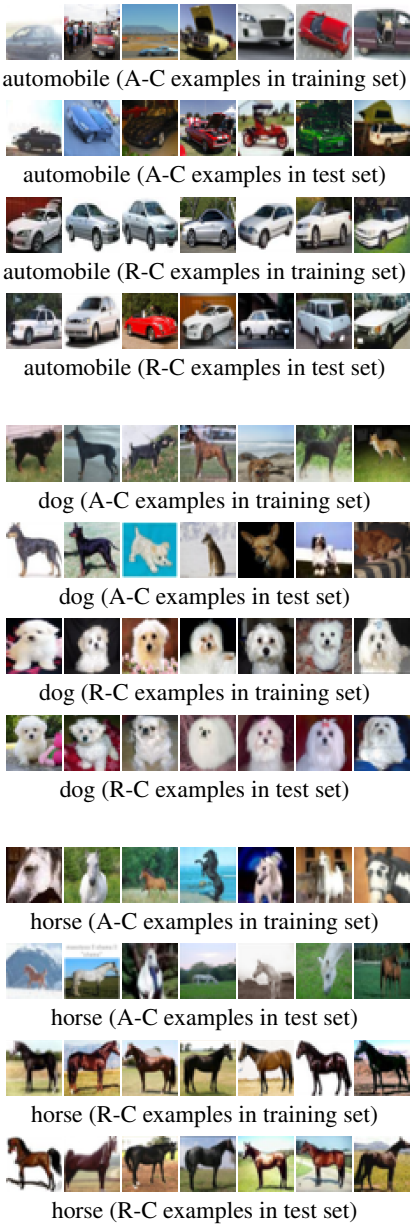


Figure 8: Illustration of A-C/R-C examples in CIFAR10 dataset.

classification. So this operation will reduce the robustness learning on R-C examples.

GAIRAT: GAIRAT reweights the PGDAT loss for each example $\boldsymbol{x}_i$ with the factor w_i^{G} calculated by the geometry value of $\boldsymbol{x}_i$. Generally, the geometry value of A-C example is smaller than R-C example. So GAIRAT will assign a larger weight for A-C example than for R-C example, which means GAIRAT focuses the accuracy/robustness learning on A-C examples. Although GAIRAT has great robustness under PGD evaluation, some works find its robustness may drop when evaluated on other attacks [4, 3].

TEAT: Like SAT, TEAT applies the temporal ensembling approach to create the preferable target vector $\boldsymbol{t}_i^{\mathrm{T}}$ instead of the original one-hot target. However, TEAT also uses $\boldsymbol{t}_i^{\mathrm{T}}$ to generate adversarial examples during training, which has a greater impact on A-C examples than on R-C examples and causes a reduction in the robustness learning on A-C examples.

B.2. Correlation with robustness condifdence

As introduced earlier for the main idea of various AT methods, our robustness condifdence and their indicators are related in design concept. Visually, we illustrate the correlation between c and two metric, learning stability [2] and geometry value [8] in Figure 9. Such positive and negative correlations demonstrate the utility of c in providing a uniform analysis of example treatments across different AT methods.

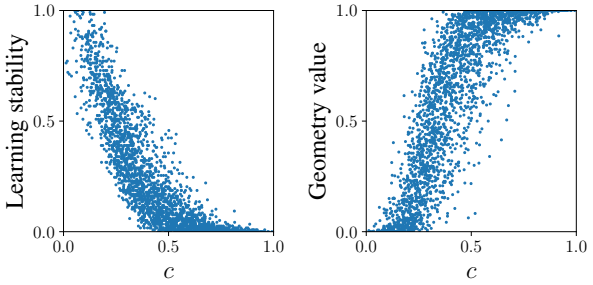


Figure 9: Correlation of robustness confidence C with other metrics: learning stability and geometry value. Every point represents a training example. We normalize the metrics to the range $[0, 1]$.

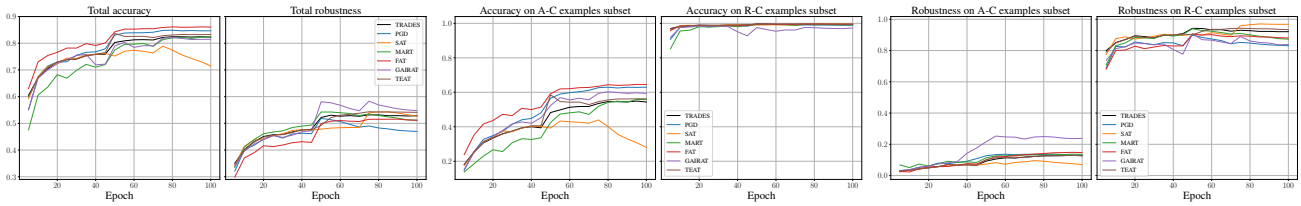
C. Performance on A-C/R-C test examples

Here we further illustrate test results on A-C and R-C examples subsets. In Figure 10, we show the results of various training methods, which can validate Table 1. Figure 11 is the supplementary results for Figure 4.

D. More about the experiment

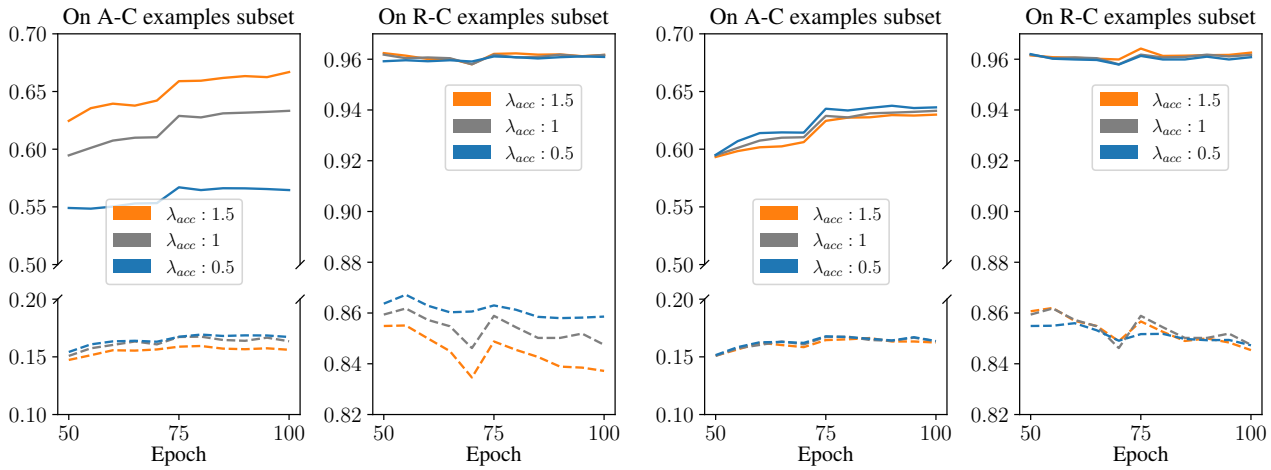
D.1. Detailed hyper-parameters

For all trials, the number of training epochs is 100. The optimizer is SGD with 0.9 momentum and 2×10^{-4} weight decay. The perturbation bound ε is 8 (pixel values in $\boldsymbol{x}$ are within the range $[0, 255]$). For multi-step AT methods TRADES and TEAT, we use the step-style scheduler to adjust the learning rate following their default settings. Specifically, the learning rate is initially set to 0.1 and decayed by 0.1 at epochs 50 and 75. The PGD iterations for training and test are 10 and 20, respectively. The PGD

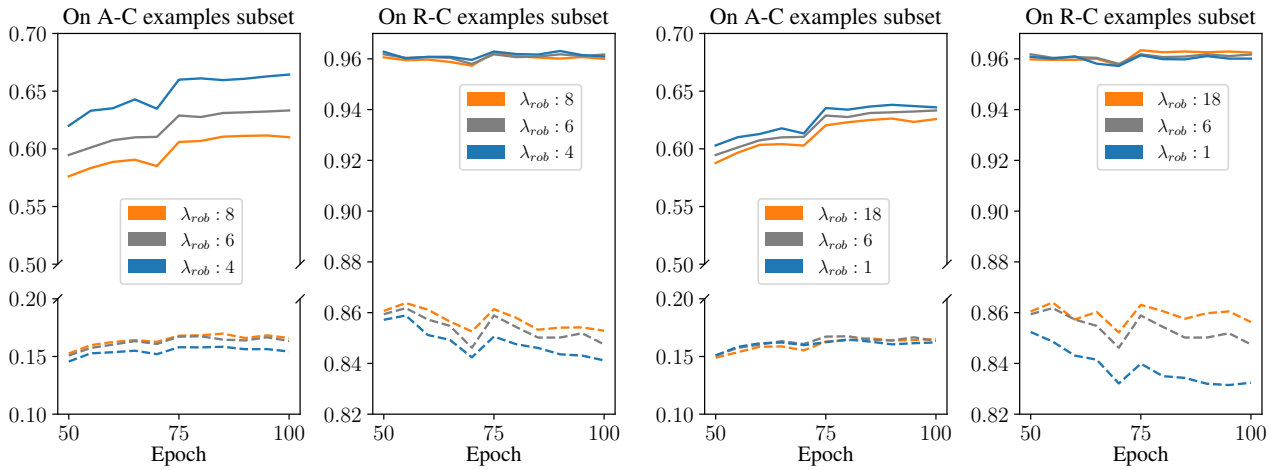


(a) Curves on whole test set (b) Accuracy curves on test subsets (c) Robustness curves on test subsets

Figure 10: Test curves of different AT methods. Dataset: CIFAR10. Model: PreActResNet-18. A-C and R-C examples subsets contain 30% of the test examples with the smallest or largest C, respectively.



(a) Change λ_{acc} on $\mathcal{S}_{AC}$ (b) Change λ_{acc} on $\mathcal{S}_{RC}$



(c) Change λ_{rob} on $\mathcal{S}_{AC}$ (d) Change λ_{rob} on $\mathcal{S}_{RC}$

Figure 11: Additional results of the experiments in Figure 4. A-C and R-C examples subsets contain 30% of the test examples with the smallest or largest C, respectively.

step-size for training and test is $2/255$. For single-step AT methods FastAT and GradAlign, we use the cyclic-style scheduler [7] with maximum learning rate 0.2 and minimum learning rate 0. The PGD iterations for training and test are 1 and 50, respectively. The PGD step-size for training and test is $10/255$ and $2/255$, respectively.

D.2. Quantitative results for trade-off comparisons

In addition to the multi-step AT method results presented in the main paper, we present the evaluation results using

AutoAttack on different datasets (including TinyImageNet) in Table 4. For the original methods, we set λ to 6 following the default settings of TRADES and TEAT. For the updated methods, we adjust $\lambda_{\min}$ and $\lambda_{\max}$ to align either the accuracy or robustness with the original method, enabling a more straightforward comparison. The results indicate an improvement in both accuracy and robustness after applying our treatment.

Table 4: Test performance (%) of original and updated (denoted with $\star$) multi-step AT methods.

Dataset	Method	λ / $\lambda_{\min}$-$\lambda_{\max}$	Acc	Rob (Auto)
CIFAR10	TRADES	6	81.12	48.17
	TRADES $\star$	4-12	**82.09**	**48.64**
	TEAT	6	82.65	48.23
	TEAT $\star$	4-10	**83.35**	**48.71**
CIFAR100	TRADES	6	54.73	23.10
	TRADES $\star$	5-8	**55.51**	**23.42**
	TEAT	6	55.82	22.54
	TEAT $\star$	3-11	**56.18**	**23.03**
TinyImageNet	TRADES	6	51.35	19.12
	TRADES $\star$	4-9	**51.40**	**20.58**
	TEAT	6	51.41	17.47
	TEAT $\star$	3-9	**51.56**	**18.12**

D.3. Ablation studies

In the application of our proposed treatment, the hyperparameters $\lambda_{\min}$ and $\lambda_{\max}$ for multi-step AT, and $a_{\min}$ and $a_{\max}$ for single-step AT, determine the contribution of A-C examples to accuracy and R-C examples to robustness. We present the results of ablation studies for these hyperparameters in Table 5 and Table 6. These results align with our earlier conclusions: smaller values of $\lambda_{\min}$/$a_{\min}$ reduce robustness learning on A-C examples, leading to improved accuracy but decreased robustness. Conversely, larger values of $\lambda_{\max}$/$a_{\max}$ enhance robustness learning on R-C examples, resulting in increased robustness but reduced accuracy. Nevertheless, our treatment effectively improve the overall trade-off between accuracy and robustness.

D.4. Fewer training epochs for single-step AT

In the main paper, we demonstrated that FastAT encounters catastrophic overfitting when trained for 100 epochs. However, if trained for fewer epochs, this issue can be mitigated. In this setting, our treatment can also improve the accuracy-robustness trade-off of FastAT, as illustrated in Figure 12. Specifically, we evaluate a values of 2, 4, 6, 8, and 10, and $(a_{\min}, a_{\max})$ pairs of $(1, 6)$, $(3, 8)$, $(5, 10)$, $(7, 12)$, and $(9, 14)$. The updated FastAT consistently achieves better either the robustness or accuracy than the original FastAT while maintaining the same level of accuracy or robustness, respectively.

Table 5: Test performance (%) of updated TRADES method with different $\lambda_{\min}$, $\lambda_{\max}$. The robustness is evaluted by PGD-20 attack.

Method	Dataset	$\lambda_{\min} - \lambda_{\max}$	Accuracy	Robustness
TRADES $\star$	CIFAR10	1-4	85.46	48.23
		1-7	84.51	49.33
		1-10	83.68	50.29
		2-10	83.47	50.92
		3-10	83.02	51.45
		4-10	82.68	51.69
	CIFAR100	1-3	59.13	24.21
		1-5	57.91	25.97
		1-7	56.72	26.94
		2-7	56.52	27.24
		3-7	55.93	27.43
		4-7	55.65	27.82

Table 6: Test performance (%) of updated FastAT method with different $a_{\min}$, $a_{\max}$. The robustness is evaluted by PGD-50 attack.

Method	Dataset	$a_{\min} - a_{\max}$	Accuracy	Robustness
FastAT $\star$	CIFAR10	1-16	90.63	37.14
		1-18	90.06	38.17
		1-20	89.89	38.66
		2-20	89.22	38.98
		3-20	88.63	39.11
		4-20	88.18	39.58
	CIFAR100	1-10	64.28	15.27
		1-12	63.64	16.55
		1-14	62.82	17.43
		2-14	62.45	17.65
		3-14	60.96	17.72
		4-14	60.53	17.81

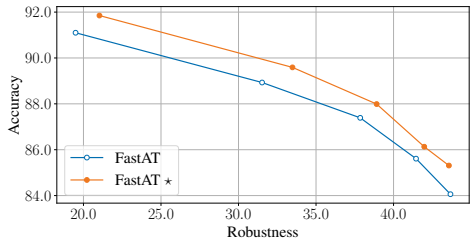


Figure 12: The trade-off comparison between original and updated (denoted with $\star$) FastAT method after 30 epochs of training on CIFAR10 dataset. The robustness is evaluated using PGD-50 attack. To achieve better accuracy or robustness, we vary the parameter a for the original FastAT method and $(a_{\min}, a_{\max})$ for our updated version.

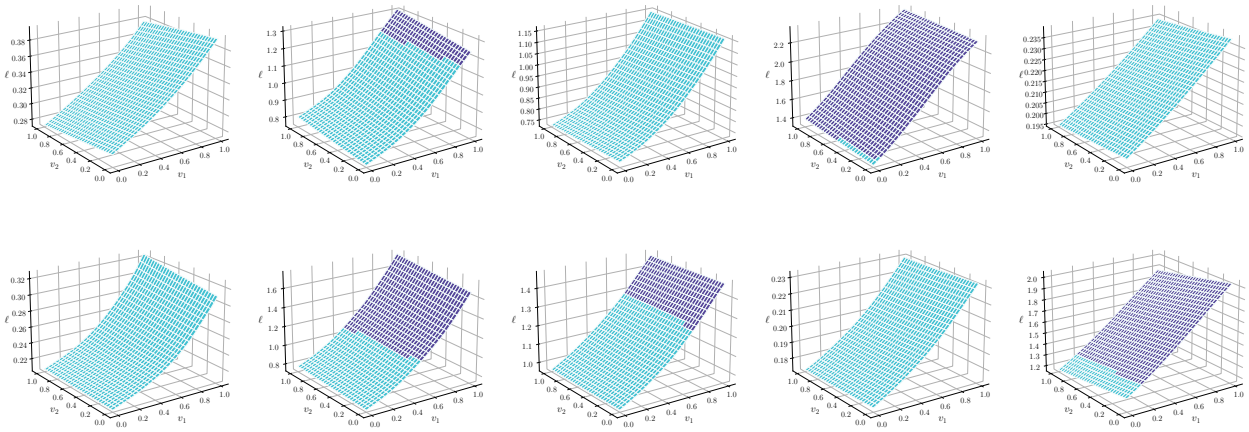


Figure 13: Loss surfaces of some test examples. ℓ is the CE loss value. v_1 is the direction of the adversarial perturbation and v_2 is the random direction. The purple (cyan) color indicates that the example at this position is misclassified (correctly classified).

Table 7: Test performance (%) of Subspace AT on CIFAR10 dataset. For the Subspace AT method, we built the subspace using 60 epochs for single-step AT and 100 epochs for multi-step AT. We then performed subspace-based training for 40 epochs.

	Subspace	Accuracy	Robustness
Multi-step	TRADES	78.11	51.63
	TRADES $\star$	**78.46**	**52.74**
Single-step	FastAT	83.18	40.05
	FastAT $\star$	**89.32**	**41.92**

D.5. Update Subspace AT

From an optimization perspective, the Subspace AT method improves both single-step and multi-step AT by constraining AT in a carefully extracted subspace [5]. We show that, when combining our treatment in building the subspace, Subspace AT can achieve better performance in both single-step and multi-step settings. The results are presented in Table 7.

E. No reliability on gradient obfuscation

We randomly select some test examples and show their loss surfaces in Figure 13. The linearity illustrated by these surfaces confirms that the robustness of models trained by RCAT is not due to gradient obfuscation [1, 6].

References

[1] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, 2018.

[2] Chengyu Dong, Liyuan Liu, and Jingbo Shang. Data quality matters for adversarial training: An empirical study. *arXiv preprint*, arXiv:2102.07437, 2021.

[3] Ruize Gao, Feng Liu, Kaiwen Zhou, Gang Niu, Bo Han, and James Cheng. Local reweighting for adversarial training. *arXiv preprint*, arXiv:2106.15776, 2021.

[4] 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.

[5] Tao Li, Yingwen Wu, Sizhe Chen, Kun Fang, and Xiaolin Huang. Subspace adversarial training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial robustness through local linearization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[7] Leslie N. Smith. Cyclical learning rates for training neural networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

[8] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *International Conference on Learning Representations (ICLR)*, 2021.