Efficient and Effective Augmentation Strategy for Adversarial Training

Sravanti Addepalli [*1] **Samyak Jain** [*1] **R.Venkatesh Babu** [1]

Abstract

The sample complexity of Adversarial training is known to be significantly higher than standard ERM based training. Although complex augmentation techniques have led to large gains in standard training, they have not been successful with Adversarial Training. In this work we propose Diverse Augmentation based Joint Adversarial Training (DAJAT), that uses a combination of simple and complex augmentations with separate batch normalization layers to handle the conflicting goals of enhancing the diversity of the training dataset and being close to the test distribution. We further introduce a Jensen-Shannon divergence loss to encourage the joint learning of the diverse augmentations, thereby allowing simple augmentations to guide the learning of complex ones. Lastly, to improve the computational efficiency of the proposed method, we propose and utilize a two-step defense, Ascending Constraint Adversarial Training (ACAT) that uses an increasing epsilon schedule and weight-space smoothing to prevent gradient masking. The proposed method achieves better performance compared to existing methods on the RobustBench Leaderboard for CIFAR-10 and CIFAR-100 on ResNet-18 and WideResNet-34-10 architectures.

1. Introduction

Deep Neural Network based classifiers are vulnerable to crafted imperceptible perturbations known as Adversarial Attacks (Szegedy et al., 2013) that can flip the predictions of the model to unrelated classes leading to disastrous implications. Adversarial Training (Goodfellow et al., 2015; Madry et al., 2018; Zhang et al., 2019) has been the most successful defense strategy, where a model is explicitly trained to be robust in the presence of such attacks. While early defenses focused on designing suitable loss functions for training, subsequent works (Pang et al., 2021; Rice et al., 2020) showed that with careful hyperparameter tuning, even the two most popular methods PGD-AT (Madry et al., 2018) and TRADES (Zhang et al., 2019) yield comparable performance, highlighting the saturation in performance with respect to changes in the training loss. Schmidt et al. (Schmidt et al., 2018) observed that adversarial training has a large sample complexity and further gains require the use of additional training data. Subsequent works (Carmon et al., 2019; Gowal et al., 2021) indeed used additional data whose distribution is close to that of the original dataset, in order to obtain performance gains. This large data requirement, which is impractical to assume, has led to an exploration towards augmentations based on Generative Adversarial Networks (Goodfellow et al., 2014) and Diffusion based models (Ho et al., 2020; Gowal et al., 2021). However, the use of such generative models incurs an additional training cost, and suffers from limited diversity in low-data regimes, and in datasets with high resolution images.

While standard Empirical Risk Minimization (ERM) based training also benefits from the use of additional data, the most practical augmentation method has been the use random transformations such as Crop, Rotation, Color Jitter, and contrast, sharpness and brightness adjustments (Krizhevsky et al., 2012; Cubuk et al., 2018; 2020). Some of these augmentations change the images significantly in input space while belonging to the same class as the original image. However, prior works (Rice et al., 2020; Gowal et al., 2020; Stutz et al., 2021) have surprisingly found that augmentations which cause large changes in the input distribution do not help adversarial training significantly. Thus, the commonly used augmentations in adversarial training are the simple transformations, zero padding followed by random crop, and horizontal flip (Rice et al., 2020; Pang et al., 2021; Gowal et al., 2020).

In this work, we show that **it is indeed possible to utilize complex augmentations effectively in Adversarial training as well**, by jointly training on simple and complex data augmentations using separate batch-normalization layers for each kind of augmentation. While complex augmentations increase the data diversity resulting in better generalization, simple augmentations ensure that the model specializes on the training data distribution as well. We

[*] Equal contribution [1] Video Analytics Lab, Department of Computational and Data Sciences, Indian Institute of Science, Bangalore. Correspondence to: Sravanti Addepalli <sravantia@iisc.ac.in>.

AdvML Frontiers workshop at 39th International Conference on Machine Learning, Baltimore, Maryland, USA, 2022. Copyright 2022 by the author(s).

further minimize the Jenson-Shannon divergence between the softmax outputs of the augmented images to enable the simple augmentations to guide the learning of complex ones. In order to improve the computational efficiency of the proposed method, we use two attack steps (instead of 10) during training. We further show that by progressively increasing the magnitude of perturbations and performing smoothing in weight space, it is indeed possible to improve the stability of training. Our contributions are listed below:

- We propose *Diverse Augmentation based Joint Adversarial Training* (DAJAT) to utilize data augmentations effectively in Adversarial training. The proposed approach can be integrated with many augmentations and adversarial training methods to improve performance.
- We propose and integrate DAJAT with an efficient 2-step defense, *Ascending Constraint Adversarial Training* (ACAT) that uses linearly increasing " schedule, cosine learning rate and weight-space smoothing to prevent gradient masking and improve convergence.
- We obtain improved robustness and large gains in standard accuracy on multiple datasets (CIFAR-10, CIFAR-100, ImageNette) and models (RN-18, WRN-34-10).
- We obtain remarkable gains in a low data scenario where data augmentations are most effective. On CIFAR-100, we outperform all existing methods on the RobustBench leaderboard (Croce et al., 2021), including the ones that utilize additional training data.

2. Diverse Augmentation based Joint Adversarial Training (DAJAT)

The use of augmentations in training can be viewed as a problem of domain generalization, where performance on the source distribution or augmented dataset is crucial towards improving the performance on the target distribution or test set. Since adversarial training is inherently challenging, for limited model capacity it is difficult to obtain good performance on the training data that is transformed using complex augmentations. Moreover, the large distribution shift between augmented data and test data, specifically with respect to low-level statistics, results in poor generalization of robust accuracy to the test set.

To mitigate these challenges, we propose the combined use of simple and complex augmentations during training so that the model can benefit from the diversity introduced by complex augmentations, while also specializing on the original data distribution that is similar to the simple augmentations. We propose to use separate batch normalization layers for simple and complex augmentations, so as to offset the shift in distribution between the two kinds of augmentations. Motivated by AugMix (Hendrycks* et al., 2020), we additionally minimize the Jenson-Shannon (JS) divergence between the softmax outputs of different augmentations, so as to allow the simple augmentations to guide the learning of complex ones. We present the training loss of the proposed Diverse Augmentation based Joint Adversarial Training (DAJAT) below:

$$L_{\mathrm{TR}} = L_{\mathrm{CE}}(f\ (x); y) + \max_{\tilde{x} 2 A_{"}(x)} \mathrm{KL}(f\ (x) jj f\ (\tilde{x})) \quad (1)$$

$$\tilde{} = \operatorname*{argmax}_{\hat{} 2 M(\ )} \frac{1}{N} \mathop{X}\limits_{i=1}^{N} L_{\mathrm{TR}}(\hat{}; x_{i;\mathrm{base}}; y_i) \quad (2)$$

$$L_{\mathrm{DAJAT}} = \frac{1}{T+1} \frac{1}{N} \mathop{X}\limits_{i=1}^{N} L_{\mathrm{TR}}(\tilde{}; x_{i;\mathrm{base}}; y_i) + \mathop{X}\limits_{t=1}^{T} L_{\mathrm{TR}}(\tilde{}; x_{i;\mathrm{auto(t)}}; y_i) + \frac{1}{N} \mathop{X}\limits_{i=1}^{N} \mathrm{JSD}(f_{\tilde{}}(x_{i;\mathrm{base}}); f_{\tilde{}}(x_{i;\mathrm{auto(1)}}); :::; f_{\tilde{}}(x_{i;\mathrm{auto(T)}})) \quad (3)$$

Adversarial attacks are generated individually for each augmentation by maximizing the respective KL divergence term of the TRADES loss shown in Eq.1. To improve training efficiency, we compute $\tilde{x}$ using two attack steps with a step-size of ". We use a combination of a linearly increasing schedule of ", cosine learning rate schedule and model weight-averaging (Izmailov et al., 2018) to improve the stability and performance of adversarial training (Details in Sec.3). The DAJAT loss (Eq.3) is a combination of the TRADES 2-step loss on each of the augmentations x_{base} and $x_{\mathrm{auto(t)}}$, along with an adversarial weight perturbation step on the loss corresponding to the base augmentations alone to reduce computational cost. For every batch normalization layer, two sets of running statistics and affine parameters are maintained and used for simple and complex augmentations respectively (Ref: Algorithm-2).

The role of the base augmentations is primarily to learn the batch normalization layers that would be used during inference time, and also to provide better supervision for the training of complex augmentations using the JS divergence term. The role of the complex augmentations is to enhance the diversity of the training dataset. Therefore we use a single base augmentation and multiple instances of a specific complex augmentation strategy such as AutoAugment (Cubuk et al., 2018). The gains in performance saturate with the addition of more complex augmentations, and therefore the use of a single base augmentation and two instances of a specific complex augmentation strategy achieves the best performance-accuracy trade-off. We note from Table-1 that in this setting, the computational complexity of the proposed method is on par with the TRADES-AWP (Wu et al., 2020) defense which is the current state-of-the-art approach.

Split Batch Normalization Layers for Different Augmentations: The proposed defense DAJAT uses separate batch normalization layers for simple and complex augmentations

Table 1. Performance of the proposed defenses ACAT and DAJAT when compared to state-of-the-art defenses on CIFAR-10, CIFAR-100 and IN-10 datasets. Robust evaluations are done against AutoAttack (AA) (Croce & Hein, 2020).

	CIFAR-10, ResNet-18				CIFAR-10, WRN-34-10		CIFAR-100, ResNet-18		CIFAR-100, WRN-34-10		IN-10, ResNet-18	
Training algorithm	Steps	Clean Acc	Robust Acc	Time/epoch (sec)	Clean Acc	Robust Acc	Clean Acc	Robust Acc	Clean Acc	Robust Acc	Clean Acc	Robust Acc
ACAT, Ours (Base, 2step)	2	82.41	49.80	95	86.71	55.36	62.05	26.10	65.75	30.23	82.34	56.96
TRADES-AWP	10	80.47	49.87	228	85.19	55.69	58.81	25.30	62.41	29.54	82.73	57.40
TRADES-AWP-WA	10	80.41	49.67	228	85.10	55.87	59.88	25.52	62.73	29.59	82.03	56.89
TRADES-AWP-WA (200 epochs)	10	81.99	51.45	228	85.36	56.17	59.11	25.97	60.30	28.68	83.41	57.91
DAJAT, Ours (Base, AA)	2 + 2	85.60	51.06	160	87.87	56.68	65.75	27.21	67.82	31.26	85.27	61.19
DAJAT, Ours (Base, 2*AA)	2 + 4	85.99	51.48	219	88.90	56.96	66.84	27.32	68.74	31.30	86.01	62.31
DAJAT, Ours (Base, 3*AA)	2 + 6	86.67	51.56	280	88.64	57.05	66.96	27.62	70.35	30.89	86.92	61.89
DAJAT, Ours (Base, 2*AA) (200 epochs)	2-5 + 4-10	85.59	52.50	293	88.71	57.81	65.45	27.69	68.75	31.85	86.26	63.21

Figure 1. Cosine Similarity of the two sets of Batch Normalization (BN) layer statistics for a WRN-34-10 model trained on CIFAR-10 using DAJAT (Base, 2*AA). BN layers corresponding to Base augmentations (Pad+Crop,H-Flip) are compared with those of AutoAugment. Parameters of initial layer (Layer-3) channels are diverse, while those of deeper layers (Layer-25) are similar.

as discussed above. A Batch Normalization (BN) layer is implemented as follows on a given feature map $g(x_i)$ of the input image x_i: $\hat{g}(x_i) = \frac{g(x_i)}{\;} \quad +$

In the proposed approach we maintain two sets of batch normalization statistics (and), and two sets of affine parameters (and) for every batch normalization layer. We plot the cosine similarity between the batch normalization vectors corresponding to the base augmentations and autoaugment of every layer in Fig.1. While the mean and variance of the batch normalization have a high similarity across all layers, we note significant differences in the and values, specifically in the initial layers. This shows that the difference in low-level statistics between the two distributions of images are being offset effectively by incorporating separate batch normalization layers. The network learns more similar parameters in deeper layers since the feature representations of different types of augmentations are expected to be more aligned in these layers.

3. Ascending Constraint Adversarial Training

In this section, we discuss the methods incorporated to improve the training efficiency of DAJAT in greater detail. We apply these methods to the TRADES-AWP defense to independently analyse their impact, and term the proposed defense as Ascending Constraint Adversarial Training (ACAT). We aim to improve the training efficiency by reducing the number of attack steps of the base defense from 10 to 2. We use two attack steps for training since it is more stable when compared to single-step adversarial training, while still being computationally efficient (Sriramanan et al., 2021).

Figure 2. Comparison of the proposed 2-step defense ACAT against TRADES-AWP (Wu et al., 2020) 2-step baseline on the CIFAR-10 with ResNet-18 architecture. ACAT has better performance and stability, especially at large training ε values. Robust Accuracy is reported against GAMA attack (Sriramanan et al., 2020) with $\varepsilon = 8/255$

As shown in Fig.2, naively reducing the number of attack steps to 2 in TRADES-AWP AT (Fixed constraint AT) causes a drop in clean and robust accuracy. While the drop is larger at higher training ε, a drop in clean accuracy is seen at $\varepsilon = 8/255$ as well. Further, the large robustness gap between last and best epochs indicates that the training stability deteriorates towards the end of training.

Prior works (Shaeiri et al., 2020) have shown that training convergence at large ℓ_1 norm bounds can be improved by linearly increasing the perturbation radius ε as training progresses. Inspired by this, we propose Ascending Constraint Adversarial Training (ACAT) that utilizes an increasing ε schedule alongside a cosine learning rate schedule with TRADES-AWP (Wu et al., 2020) loss formulation for improving the stability and convergence of two-step adversarial training. We use a cosine learning rate schedule that decays monotonically over the training epochs, since at large training ε, lower learning rate could further stabilize training. As shown in Fig.2, the performance and stability of the proposed 2-step defense ACAT are better when compared to the TRADES-AWP 2-step baseline, at the same computational cost, specifically at larger perturbation bounds of $12/255$ and $16/255$. The proposed defense maintains a good clean accuracy at all the training ε values considered, and has almost 0 difference between best and last epochs.

4. Experiments and Results

We compare the proposed approach against several state-of-the-art defenses in Table-1 on CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009) and ImageNette (Howard & Gugger, 2020). We integrate model weight averaging with the TRADES-AWP baseline as well (TRADES-AWP-WA).

Firstly, we compare the proposed 2-step defense ACAT with the existing state-of-the-art 2-step defense NuAT-WA (Sriramanan et al., 2021) and GAT (Sriramanan et al., 2020) in Table-2. We obtain a marginal boost in both clean and robust accuracy on WideResNet-34-10. Moreover, ACAT can be integrated with the Nuclear Norm training objective as well to obtain improved results. The performance of the proposed ACAT defense is superior when compared to the multi-step training method PGD-AT (Rice et al., 2020) as well. When compared to the TRADES-AWP 10-step defense (Wu et al., 2020; Zhang et al., 2019), we obtain improved clean accuracy with a slight drop in robust accuracy at half the computational cost. On the CIFAR-100 dataset, we obtain substantial gains in both clean and robust accuracy when compared to the 10-step baselines.

We present three variants of the proposed defense DAJAT by using one, two and three AutoAugment based augmentations for every image. We denote them as DAJAT(Base, AA), DAJAT(Base, 2*AA) and DAJAT(Base, 3*AA) respectively. Using a single AutoAugment based augmentaion (Base, AA), we obtain improved clean and robust accuracy when compared to most of the baselines considered across all datasets and models. By increasing the number of AutoAugment based transformations to 2, we observe consistent gains in robust and clean accuracy in all cases. In this setting, the computational complexity of the proposed approach matches with that of TRADES-AWP (Wu et al., 2020) as shown in Table-1. With the setting (Base, 3*AA), we obtain marginal improvements in performance.

Overall, using the (Base, 2*AA) approach, which has comparable time complexity as the TRADES-AWP 10-step defense, we obtain large gains ranging from 3.8% to 7% on clean accuracy and around 1.8% higher robust accuracy against AutoAttack (Croce & Hein, 2020) across most settings. On the Imagenette dataset (Howard & Gugger, 2020) we obtain 4.2% higher clean accuracy and 4.49% higher robust accuracy, showing that augmentation strategies work best when the amount of training data is less when compared to the complexity of the task. While the use of 2 attack steps helps in improving the training efficiency of DAJAT, we show that by using more attack steps and longer training epochs, we can indeed obtain further gains in performance. The varying" schedule in DAJAT allows the use of an increasing schedule in the number of steps as well, thereby limiting the overall cost associated with higher attack steps. We present results by increasing the number of attack steps from 2 to 5 uniformly every 50 epochs in last row of Table-1.

4.1. Combining DAJAT with other augmentations

We explore combining the proposed defense DAJAT with other augmentations in Table-5. We do not use the JS divergence term for Cutmix and Mixup since they involve changes in the label space. We note that without using any augmentation in the training dataset, we obtain poor clean and robust accuracy, highlighting the importance of using Pad and Crop followed by horizontal flip. The proposed approach is able to obtain good performance gains using AutoAugment (Cubuk et al., 2018), Color Jitter and CutOut (DeVries & Taylor, 2017) augmentations, highlighting that it can work well with pixel-level and spatial augmentations.

The use of Cutmix and Mixup in adversarial training are challenging since they involve changes in label space. Although the base accuracy using these augmentations is poor, using DAJAT we obtain considerable gains, highlighting that it enables the use of a variety of augmentations without the need for careful selection. We note that Rebuffi et al. (Rebuffi et al., 2021) obtain considerable gains using Cutmix along with many other improvements. By incorporating some of the tricks reproduced by Rade et al. (Rade, 2021) we obtain improvements in the CutMix baseline and further gains in the proposed method. For a similar computational budget, we compare with Rebuffi et al. (Rebuffi et al., 2021) in Table-6, where we obtain gains of 1.7% clean and 3.3% robust accuracy (Details in Sec.C.3). We present ablation results for the proposed approach in Table-4.

5. Conclusions

Contrary to prior knowledge, we show that it is indeed possible to use common augmentation strategies that modify the low-level statistics of images to improve the performance of adversarial training. We propose a novel defense Diverse Augmentation based Joint Adversarial Training (DAJAT) that uses a combination of simple and complex augmentations with separate batch normalization layers to allow the network training to benefit from the diverse training data distribution obtained using complex augmentations, while also being trained on a distribution that is close to the test set. The use of JS divergence term between network predictions of different augmentations enables the joint learning across various augmentations. We improve the efficiency of DAJAT by utilizing the proposed two-step defense strategy Ascending Constraint Adversarial Training (ACAT) that improves the stability and performance of TRADES 2-step adversarial training significantly by using a linearly increasing " schedule along with a cosine learning rate schedule and weight-space smoothing.

6. Acknowledgements

This work was supported by a research grant (CRG/2021/005925) from SERB, DST, Govt. of India. Sravanti Addepalli is supported by Google PhD Fellowship and CII-SERB Prime Minister's Fellowship for Doctoral Research. We are thankful for the support.

References

Addepalli, S., Jain, S., Sriramanan, G., Khare, S., and Radhakrishnan, V. B. Towards achieving adversarial robustness beyond perceptual limits. In ICML 2021 Workshop on Adversarial Machine Learning, 2021. URL https://openreview.net/forum?id=SHB_znlW5G7.

Andriushchenko, M., Croce, F., Flammarion, N., and Hein, M. Square attack: a query-efficient black-box adversarial attack via random search. In The European Conference on Computer Vision (ECCV), 2020.

Athalye, A., Carlini, N., and Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In International Conference on Machine Learning (ICML), 2018.

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. S. Unlabeled data improves adversarial robustness. In Advances in Neural Information Processing Systems (NeurIPS), 2019.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In International Conference on Machine Learning (ICML), 2020.

Croce, F., Andriushchenko, M., Sehwag, V., Debenedetti, E., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark, 2021.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501, 2018.

Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 702–703, 2020.

DeVries, T. and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552, 2017.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In International Conference on Learning Representations (ICLR), 2015.

Gowal, S., Qin, C., Uesato, J., Mann, T., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples. arXiv preprint arXiv:2010.03593, 2020.

Gowal, S., Rebuffi, S.-A., Wiles, O., Stimberg, F., Calian, D. A., and Mann, T. A. Improving robustness using generated data. Advances in Neural Information Processing Systems, 34, 2021.

Hendrycks*, D., Mu*, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. Augmix: A simple method to improve robustness and uncertainty under data shift. In International Conference on Learning Representations, 2020. URL https://openreview.net/forum?id=S1gmrxHFvB.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. Advances in Neural Information Processing Systems, 33:6840–6851, 2020.

Howard, J. and Gugger, S. Fastai: a layered api for deep learning. Information, 11(2):108, 2020.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. P., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. ArXiv, abs/1803.05407, 2018.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 2012.

Krizhevsky, A. et al. Learning multiple layers of features from tiny images. 2009.

Madry, A., Makelov, A., Schmidt, L., Dimitris, T., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations (ICLR), 2018.

Pang, T., Yang, X., Dong, Y., Su, H., and Zhu, J. Bag of tricks for adversarial training. In International Conference on Learning Representations (ICLR), 2021.

Rade, R. PyTorch implementation of uncovering the limits of adversarial training against norm-bounded adversarial examples, 2021. URL https://github.com/imrahulr/adversarial_robustness_pytorch.

Rade, R. and Moosavi-Dezfooli, S.-M. Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=Azh9QBQ4tR7.

Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T. A. Data augmentation can improve robustness. Advances in Neural Information Processing Systems, 34, 2021.

Rice, L., Wong, E., and Kolter, J. Z. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning (ICML)*, 2020.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. *Advances in neural information processing systems*, 31, 2018.

Shaeiri, A., Nobahari, R., and Rohban, M. H. Towards deep learning models resistant to large perturbations. *arXiv preprint arXiv:2003.13370*, 2020.

Sriramanan, G., Addepalli, S., Baburaj, A., and Venkatesh Babu, R. Guided Adversarial Attack for Evaluating and Enhancing Adversarial Defenses. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Sriramanan, G., Addepalli, S., Baburaj, A., and Venkatesh Babu, R. Towards Efficient and Effective Adversarial Training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Stutz, D., Hein, M., and Schiele, B. Relating adversarially robust generalization to flat minima. *arXiv preprint arXiv:2104.04448*, 2021.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2013.

Wu, D., Xia, S.-T., and Wang, Y. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2019.

A. Details on Ascending Constraint Adversarial Training (ACAT)

In this section, we present details on the proposed 2-step efficient defense strategy, Ascending Constraint Adversarial Training (ACAT).

A.1. ACAT training algorithm

The algorithm for the proposed ACAT defense is presented in Algorithm-1. We consider an ℓ_∞ threat model of perturbation radius ε. The perturbation bound for attack generation ε_{asc} is linearly increased from 0 to ε during training (L3). The learning rate follows a cosine schedule across the training epochs as shown in L4. The attack generation (L7-L12) is done for 2 iterations and follows the TRADES (Zhang et al., 2019) settings. Initially, Gaussian noise of magnitude 0.001 is added to every pixel (L8). The KL divergence loss between the clean and perturbed images is maximized using the perturbation step size ε_{asc} (L9). Further, the perturbation is clipped to remain within the threat model in every iteration (L10-L11).

As shown in L14, the TRADES-AWP (Wu et al., 2020) loss is used for adversarial training. The loss $L_{AWP}(\theta)$ is maximized with respect to θ to find the perturbed model weights $\tilde{\theta}$ within the constraint set $M(\theta)$ (L15). Further, the model training is done at $\tilde{\theta}$, after which the adversarial weight pertubation ζ_{AWP} is subtracted to offset the perturbation in weights (L17). The defense ACAT does not use any additional training hyperparameters when compared to the TRADES-AWP defense. We vary the hyperparameter β to obtain optimal results.

Algorithm 1 Ascending Constraint Adversarial Training (ACAT)

```
1: Input: Network $f_\theta$, Training Dataset $D = \{(x_i, y_i)\}$, Adversarial Threat model: $\ell_\infty$ bound of radius $\varepsilon$, number
   of epochs E, Maximum Learning Rate $LR_{max}$, M training mini-batches of size n, Cross-entropy loss $\ell_{CE}$, Weight
   perturbation constraint $M(\theta)$, coefficient of KL divergence term $\beta$
2: for epoch = 1 to E do
3:   $\varepsilon_{asc} = epoch \cdot \varepsilon / E$
4:   $LR = 0.5 \cdot LR_{max} \cdot (1 + cosine((epoch - 1)/E \cdot \pi))$
5:   for iter = 1 to M do
6:     for i = 1 to n (in parallel) do
7:       for steps = 1 to 2 do
8:         $\delta = 0.001 \cdot N(0, 1)$
9:         $\delta = \delta + \varepsilon_{asc} \cdot sign(\nabla_\delta KL(f_\theta(x_i) || f_\theta(x_i + \delta)))$
10:        $\delta = Clamp(\delta, -\varepsilon_{asc}, \varepsilon_{asc})$
11:        $\tilde{x}_i = Clamp(x_i + \delta, 0, 1)$
12:      end for
13:    end for
14:    $L_{AWP}(\theta) = \frac{1}{n} \sum_{i=1}^{n} L_{CE}(f_\theta(x_i), y_i) + \beta \cdot KL(f_\theta(x_i) || f_\theta(\tilde{x}_i))$
15:    $\tilde{\theta} = \mathrm{argmax}_{\theta \in M(\theta)} L_{AWP}(\theta)$
16:    $\zeta_{AWP} = \tilde{\theta} - \theta$
17:    $\theta = \tilde{\theta} - LR \cdot \nabla_{\tilde{\theta}}(L_{AWP}(\tilde{\theta})) - \zeta_{AWP}$
18:  end for
19: end for
```

A.2. Integrating ACAT with other efficient training methods

The proposed ACAT defense uses the KL divergence loss between clean and adversarial images, similar to the TRADES adversarial training algorithm (Zhang et al., 2019). We present results by integrating the proposed ACAT defense with losses from existing efficient adversarial training algorithms (Sriramanan et al., 2020; 2021) in Table-2. We obtain a significant boost in performance over the respective baselines, when we use ACAT with GAT (Sriramanan et al., 2020) and TRADES (Zhang et al., 2019) losses, and a marginal boost when integrated with the NuAT defense (Sriramanan et al., 2021). The adversarial weight perturbation step in the proposed defense results in an increase in computational time when compared to the respective baselines. We choose the KL divergence based loss for both proposed defenses ACAT and DAJAT since it results in an optimal trade-off between performance and training time.

Table 2.Integrating ACAT with different loss formulations on the CIFAR-10 dataset with WideResNet-34-10 architecture. Robust accuracy is reported against the GAMA attack (Sriramanan et al., 2020).

	# Attack Steps	Clean Acc	Robust Acc	Time per epoch (seconds)
TRADES-AWP	2	85.49	41.62	412
ACAT (with TRADES loss)	2	86.71	55.58	412
NuAT2-WA	2	86.32	55.08	334
ACAT (with NuAT loss)	2	86.19	55.91	530
GAT2-WA	2	87.36	50.24	267
ACAT (with GAT loss)	2	87.79	54.70	396

Table 3.Performance (%) of DAJAT whencombined with other Adversarial training methods, OAAT (Addepalli et al., 2021) and HAT (Rade & Moosavi-Dezfooli, 2022) onCIFAR-10 and CIFAR-100 with 110 epochs of training. Robust evaluations are performed on Auto-Attack(AA) (Croce & Hein, 2020) at $"t = 8=255$ and $16=255$.

	CIFAR-10, ResNet-18			CIFAR-10, WRN-34-10			CIFAR-100, ResNet-18			CIFAR-100, WRN-34-10		
Method	Clean	AA, 8/255	AA, 16/255	Clean	AA, 8/255	AA, 16/255	Clean	AA, 8/255	AA, 16/255	Clean	AA, 8/255	AA, 16/255
AWP (Zhang et al., 2019; Wu et al., 2020)	80.47	49.87	19.23	85.10	55.87	23.27	59.88	25.81	8.28	62.73	29.59	11.04
AWP+DAJAT	85.99	51.48	16.33	88.90	56.96	19.73	66.84	27.32	8.97	68.74	31.30	9.91
OAAT (Addepalli et al., 2021)	80.24	50.88	22.05	85.67	55.93	24.05	61.70	26.77	9.91	65.73	30.35	12.01
OAAT+DAJAT	82.05	52.21	22.78	86.22	57.64	24.56	62.50	28.47	10.67	66.03	31.15	12.67
HAT (Rade & Moosavi-Dezfooli, 2022)	85.63	49.54	14.96	86.21	51.46	16.76	59.19	23.26	6.96	59.95	24.55	7.13
HAT+DAJAT	86.68	51.47	16.38	86.71	53.85	16.50	62.78	26.49	8.72	64.88	27.37	8.71

B. Diverse Augmentation based Joint Adversarial Training (DAJAT)

B.1. Details on the proposed defense DAJAT

The algorithm of the proposed approach is presented in Algorithm-2. In every training iteration, multiple augmentations are considered for every image x_i (L7). We consider one base augmentation and T complex augmentations. The base augmentation consists of Pad and Crop followed by Horizontal Flip, while the complex augmentations are a combination of AutoAugment (Cubuk et al., 2018) and the base augmentations. The attack generation for each augmentation (L8-L13) is similar to the ACAT algorithm discussed in Section-A.1. The DAJAT loss (L16) is a combination of the TRADES loss (Zhang et al., 2019) (L17) on each augmentation, and a Jensen-Shannon (JS) divergence term between all augmentations. The JS divergence is a combination of KL divergence terms with respect to the average probability vector as shown below.

$$\mathrm{JSD}(f\ (x_{i;\,\mathrm{base}}); f\ (x_{i;\,\mathrm{auto}(1)}); :::; f\ (x_{i;\,\mathrm{auto}(T)})) = \frac{1}{T+1}\ \mathrm{KL}(f\ (x_{i;\,\mathrm{base}}; M) + \mathrm{KL}(f\ (x_{i;\,\mathrm{auto}(1)}; M) + \quad + \mathrm{KL}(f\ (x_{i;\,\mathrm{auto}(T)}; M) \qquad (4)$$

where M is defined as below,

$$M = \frac{1}{T+1}\ f\ (x_{i;\,\mathrm{base}}) + f\ (x_{i;\,\mathrm{auto}(1)}) + \quad + f\ (x_{i;\,\mathrm{auto}(T)}) \qquad (5)$$

The JS-divergence term improves accuracy on clean samples and training convergence by enabling the joint learning of representations across different augmentations. The model weights are perturbed to $\tilde{}$ by maximizing the TRADES loss on the base augmentations alone within the constraint set $M($) (L18). This constraint set is chosen such that $jj\tilde{}\ jj \quad jj\ _l jj$ for any layer l. The network $\tilde{}$ is updated using an SGD step to minimize the overall loss $L_{\mathrm{DAJAT}}\ (\tilde{})$ (L20). The model weights are further offset by $_{\mathrm{AWP}}$ which is the adversarial weight perturbation at

C. Details on Experiments and Results

C.1. Combining the proposed approach with different adversarial training methods

We explore combining the proposed defense DAJAT with some existing methods in Table-3. We observe that combining DAJAT with all three existing works (Wu et al., 2020; Addepalli et al., 2021; Rade & Moosavi-Dezfooli, 2022) leads to significant gains both in clean as well as adversarial accuracies (AA, 8/255), especially on CIFAR-100 where the number of images per class is low. Although OAAT (Addepalli et al., 2021) shows improved results over AWP (Wu et al., 2020), combining DAJAT with OAAT leads to further gains of $1:5\%$ in both clean and adversarial accuracy on CIFAR10 and $1 \quad 1:5\%$ gains in both clean and adversarial accuracy on CIFAR100. Further, since OAAT (Addepalli et al., 2021) claims

Algorithm 2 Diverse Augmentation based Joint Adversarial Training (DAJAT)

```
Input: Network $f_\theta$, Training Dataset $D = \{(x_i, y_i)\}$, Adversarial Threat model: $\ell_\infty$ bound of radius $\varepsilon$, number of epochs $E$, Maximum Learning Rate $LR_{max}$, $M$ training mini-batches of size $n$, number of attack steps $S$, Cross-entropy loss $\ell_{CE}$, Weight perturbation constraint $M(\theta)$, Number of augmented images using autoaugment $T$, coefficient of KL divergence term $\beta$
for $epoch = 1$ to $E$ do
  $\varepsilon_{asc} = epoch \cdot \varepsilon / E$
  $LR = 0.5 \cdot LR_{max} \cdot (1 + cosine((epoch - 1)/E \cdot \pi))$
  for $iter = 1$ to $M$ do
    for $i = 1$ to $n$ (in parallel) do
      for $a \in \{base, auto(1), \dots, auto(T)\}$ do
        for $steps = 1$ to $S$ do
          $\delta = 0.001 \cdot \mathcal{N}(0, 1)$
          $\delta = \delta + \varepsilon_{asc} \cdot sign(\nabla_\delta KL(f_\theta(x_{i;a}) || f_\theta(x_{i;a} + \delta)))$
          $\delta = Clamp(\delta, -\varepsilon_{asc}, \varepsilon_{asc})$
          $\tilde{x}_{i;a} = Clamp(x_{i;a} + \delta, 0, 1)$
        end for
      end for
    end for
    $L_{DAJAT}(\theta) = \frac{1}{T+1}\left[\frac{1}{n}\sum_{i=1}^{n} L_{TR}(\theta; (x_i, \tilde{x}_i)_{base}; y_i) + \sum_{t=1}^{T} L_{TR}(\theta; (x_i, \tilde{x}_i)_{auto}; y_i)\right] + \frac{1}{n}\sum_{i=1}^{n} JSD(f_\theta(x_{i;\, base}); f_\theta(x_{i;\, auto(1)}); \dots; f_\theta(x_{i;\, auto(T)}))$
    where, $L_{TR}(\theta; (x, \tilde{x}); y) = L_{CE}(f_\theta(x); y) + \beta \cdot KL(f_\theta(x) || f_\theta(\tilde{x}))$
    $\tilde{\theta} = \underset{\tilde{\theta} \in M(\theta)}{argmax} \frac{1}{n}\sum_{i=1}^{n} L_{TR}(\tilde{\theta}; (x_i, \tilde{x}_i)_{base}; y_i)$
    $\theta_{AWP} = \tilde{\theta} - \theta$
    $\theta = \tilde{\theta} - LR \cdot \nabla_{\tilde{\theta}}(L_{DAJAT}(\tilde{\theta})) - \theta_{AWP}$
  end for
end for
```

to achieve robustness at larger epsilon bounds, we evaluate using Auto-Attack at $\varepsilon = 16/255$. Using OAAT+DAJAT we observe gains over OAAT on AutoAttack with $\varepsilon = 16/255$ as well, which further confirms the effectiveness of DAJAT. Finally we combine DAJAT with HAT (Rade & Moosavi-Dezfooli, 2022) and we observe consistent gains over HAT (Rade & Moosavi-Dezfooli, 2022) on all models and datasets. While HAT proposes to improve the robustness-accuracy trade-off, combining DAJAT with HAT further improves this trade-off and shows gains of $\sim 1\%$ on clean accuracy and 2% on robust accuracy for CIFAR-10, and $3-5\%$ on clean accuracy and 3% on robust accuracy for CIFAR-100 dataset.

C.2. Ablation experiments

We present ablation experiments to highlight the significance of different components of the proposed approach in Table-4 on the CIFAR-10 dataset using ResNet-18 architecture. All experiments are run for 110 training epochs, except A7 which is run for 220 epochs. We show the importance of the JS divergence term in the proposed loss in the ablations A1-A6 of Table-4. Using the JS divergence term we obtain $\sim 1\%$ higher clean accuracy across (Base, AA), (Base, 2*AA) and (Base, 3*AA) settings of the proposed defense. For (Base, 2*AA) and (Base, 3*AA) we obtain marginal improvements in robust accuracy as well. From A7, A9 and A10, we find that the proposed JS divergence term helps even in the case where both augmentations of an image are generated using the same pipeline. Using two AutoAugment based transformations, we obtain 1.6% higher clean accuracy when compared to the 220 epoch 2-step defense at a comparable computational cost. Comparing A9, A10 and A11, we note that the use of simple and complex augmentations indeed shows improvements over the case of using 2 complex or simple augmentations alone. The importance of split-batch norm in the proposed approach can be evidently seen by comparing A12 and A14. By using single batch norm (A12), robust accuracy drops by 8.24%. Further, in this case if the JS term is also dropped, the robustness of the network is almost completely lost. This shows that

Table 4. Ablation experiments performed on the CIFAR-10 dataset using ResNet-18 architecture. Robust Accuracy is reported against GAMA attack (Sriramanan et al., 2020).

	# Steps	Clean Acc	Robust Acc		# Steps	Clean Acc	Robust Acc
[A1] Ours (Base, AA) without JS	2 + 2	84.55	51.45	[A8] Ours (Base, 2step) 110 epochs	2	82.41	50.00
[A2] Ours (Base, AA)	2 + 2	85.60	51.27	[A9] Ours (AA, AA)	2 + 2	84.68	49.71
[A3] Ours (Base, 2*AA) without JS	2 + 4	85.07	51.53	[A10] Ours (Base, Base)	2 + 2	83.93	49.88
[A4] Ours (Base, 2*AA)	2 + 4	85.99	51.71	[A11] Ours (Base, AA)	2 + 2	85.60	51.27
[A5] Ours (Base, 3*AA) without JS	2 + 6	85.31	51.67	[A12] Ours (Base, 3*AA) Single Batch Norm	2 + 6	86.68	43.57
[A6] Ours (Base, 3*AA)	2 + 6	86.67	51.81	[A13] Ours (Base, 3*AA) Single Batch Norm without JS	2 + 6	75.64	4.20
[A7] Ours (Base, 2step) 220 epochs	2	83.05	50.31	[A14] Ours (Base, 3*AA)	2 + 6	86.67	51.81

Table 5. Impact of using other augmentations in DAJAT. Performance on CIFAR-10 dataset with ResNet-18 architecture is reported. Robust evaluations are done against GAMA attack (Sriramanan et al., 2020). ^{y}PreAct-ResNet18 with Swish activation is used (Rebuf et al., 2021; Rade, 2021).

	Augmentation		Base + Aug		Base + 2 * (Aug)			Augmentation		Base + Aug		Base + 2 * (Aug)	
Augmentation	Clean	Robust	Clean	Robust	Clean	Robust	Augmentation	Clean	Robust	Clean	Robust	Clean	Robust
No Augmentation	76.32	43.20	78.08	41.71	77.42	41.07	Cutout (DeVries & Taylor, 2017)	82.38	50.14	84.91	51.40	85.11	51.60
Pad+Crop+H-Flip	82.41	50.00	83.69	51.30	83.62	51.09	Colour Jitter	82.98	48.82	84.50	51.19	84.85	51.62
AutoAugment (Cubuk et al., 2018)	82.54	48.11	84.94	51.23	85.99	51.71	Mixup (Zhang et al., 2017)	79.08	45.07	85.18	50.18	84.24	50.01
Cutmix (Yun et al., 2019)	79.03	41.57	82.33	50.90	81.64	49.50	RandAugment (Cubuk et al., 2020)	82.48	44.66	84.61	51.01	85.47	51.33
Cutmix^{y} (Rebuf et al., 2021; Rade, 2021)	82.01	47.65	84.58	50.97	85.49	51.58	Augmix (Hendrycks* et al., 2020)	82.38	48.84	84.96	50.4	85.18	50.51

using a single batch norm layer for diverse augmentations makes it harder for the network to converge. We also note that the JS divergence term indeed helps in improving the convergence of training in addition to improving performance.

C.3. Comparison against CutMix based augmentation

While we compare the performance of the proposed approach against various base adversarial training algorithms (Wu et al., 2020; Madry et al., 2018; Pang et al., 2021; Sriramanan et al., 2021) in the main paper, we additionally compare with a recent augmentation scheme that uses CutMix augmentations (Rebuf et al., 2021) to improve performance in this section. The authors (Rebuf et al., 2021) show a significant boost in performance using 400 epochs of training and large model architectures. However, to ensure a fair comparison, we report the result of 110 epochs of training on WideResNet-34-10 architecture and CIFAR-10 dataset that has been shared by the authors with us upon request. We report the PGD 40-step accuracy as shared by the authors. As shown in Table-6, we obtain a significant boost in performance over the CutMix based augmentation as well as the TRADES-AWP (Wu et al., 2020) baseline using the proposed defense DAJAT.

Additionally, contrary to the claims by Rebuf et al. (Rebuf et al., 2021), we show that it is indeed possible to effectively use augmentations that modify the low-level statistics of images for obtaining improved performance in Adversarial Training by using the proposed defense DAJAT.

As noted in the github repository[1] by Rebuf et al. (2021), we find that naively using cutmix does not give good results as shown in Table-5. Therefore, as referenced by the authors, we use the repository by Rade (2021) as the base code and incorporate cutmix into it. We present the results for 200 epochs training with learning rate drop of 0.1 at 100 and 150 epochs, using the PreActResNet-18 model with Swish Activation and batch size of 128 in Table-7(C1). We observe significantly improved results as compared to Table-5 on using the repository by Rade (2021) as the base code. We observe that the key differences in the repository by Rade (2021) as compared to the TRADES repository (Zhang et al., 2019) are:

- Use of swish activation function in the PreActResNet18 model
- Weight decay not used for batch normalization layers

To study the impact of these changes, we investigate the use ReLU instead of Swish activation (Table-7(C5)) and the use of weight decay for all the parameters of the model including the batch normalization layers (Table-7(C6)). In both cases, we observe a significant drop with respect to C1. Thus based on this ablation, the use of swish activation, and avoiding weight decay for batch normalization layers seems to be important to obtain performance gains using Cutmix.

By using linearly increasing varying epsilon schedule along with cosine learning rate, we obtain further improvements in

[1] https://github.com/deepmind/deepmind-research/tree/master/adversarialrobustness/pytorch

Table 6. Comparison of the proposed augmentation scheme with CutMix based augmentations (Rebuffi et al., 2021) Performance (%) of the proposed defense DAJAT (Base, 2*AA) when compared to the use of CutMix based augmentation proposed by Rebuffi et al. (Rebuffi et al., 2021) against PGD 40-step attack (Madry et al., 2018).

Method	Clean Acc	Robust Accuracy (PGD-40)
TRADES (Zhang et al., 2019)	84.72	56.92
Rebuffi et al. (Rebuffi et al., 2021)	87.24	57.60
TRADES-AWP (Wu et al., 2020)	85.35	59.13
Ours-DAJAT (Base, 2*AA)	88.90	60.97

Table 7. Performance (%) by using cutmix augmentations in the repository by Rade (2021) on CIFAR-10 dataset with Preact-ResNet18 model and Swish Activation. The models are trained using varying epsilon schedule and cosine learning rate unless specified otherwise. Robust Accuracy is reported against GAMA attack (Sriramanan et al., 2020).

Method	Clean Accuracy	Robust Accuracy
[C1]: TRADES + Cutmix (step LR schedule + fixed)	81.67	49.18
[C2]: TRADES + Cutmix (cosine LR schedule + varying	83.34	49.24
[C3]: Ours (Base, Cutmix)	82.67	51.99
[C4]: Ours (Base, 2*Cutmix)	83.05	52.22
[C5]: C1 + ReLU activation	81.03	46.60
[C6]: C1 + Weight decay for BN	70.66	36.36

performance as shown in C2. Next we incorporate the proposed method DAJAT and present the results in Table-7(C3,C4), where we obtain significant gains in performance over C1, thus showing the effectiveness of DAJAT.

C.4. Sanity checks to verify the absence of gradient masking

We perform several sanity checks as recommended by Athalye et al. (2018) to ensure the absence of gradient masking in the proposed defenses ACAT and DAJAT.

- From Table-8 we note that Black-Box attacks are weaker than White-Box attacks, indicating that the gradients from the model are reliable.
- We further note from Table-8 that attacks with higher number of steps are stronger than those with lower steps. Further, PGD multi-step attacks are stronger than FGSM white-box attacks.
- From Table-9 we note that robust accuracy against targeted and untargeted attacks saturates as the number of attack steps increase from 500 to 1000, indicating that the evaluation is robust.

Figure 3. Robust Accuracy and Loss against variation in perturbation size: (a,c) Robust accuracy (%) of the proposed defenses ACAT and DAJAT against PGD 7-step attacks across variation in attack perturbation bound. Attacks within larger perturbation bounds are able to bring down the robust accuracy of the model to 0, indicating the absence of gradient masking. (b,d) Cross-entropy loss on FGSM adversarial samples across variation in attack perturbation bound. The linearly increasing trend of loss indicates the absence of gradient masking. The models are trained on CIFAR-10 dataset using ResNet-18 architecture.

Table 8. **Evaluation against Black-Box (BB) and White-Box (WB) FGSM (Goodfellow et al., 2015) attacks and multi-step PGD attacks (Madry et al., 2018).** Performance (%) of the proposed defense DAJAT (Base + 2*AA) is compared against baselines on CIFAR-10 dataset with ResNet-18 architecture.

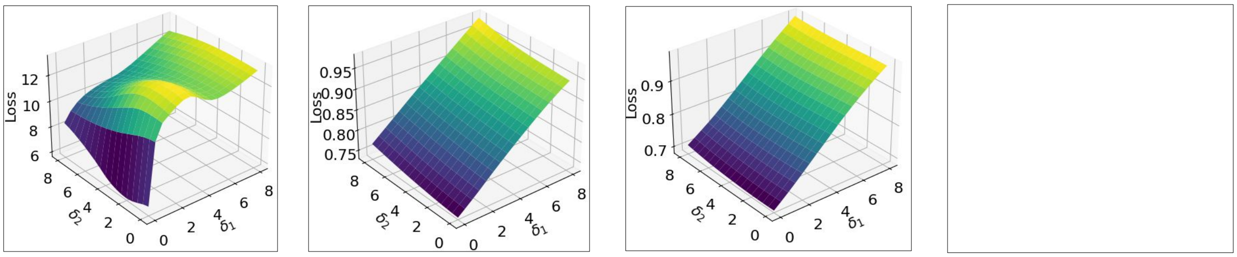
	Clean Acc	BB FGSM	WB FGSM	PGD-20	PGD-100	PGD-500
NuAT2-WA	86.32	84.71	63.48	58.09	57.74	57.74
ACAT	86.71	85.29	64.08	58.76	58.64	58.53
TRADES-AWP	85.36	83.93	63.49	59.22	59.11	59.08
DAJAT(Base, 3*AA)	**88.64**	**87.19**	**66.99**	**61.09**	**60.80**	**60.74**

Table 9. **Evaluation against multi-step Targeted and Untargeted PGD attacks (Madry et al., 2018) with single and multiple random restarts.** Performance (%) of the proposed defense DAJAT (Base, 2*AA) across different datasets with ResNet-18 architecture.

	CIFAR-10		**CIFAR-100**		**IN-10**	
Attack	500-step	1000-step	500-step	1000-step	500-step	1000-step
PGD-Targeted (Least Likely Class)	85.01	85.01	66.02	65.98	85.06	85.01
PGD-Targeted (Random Class)	80.56	80.55	63.96	63.96	80.13	80.13
PGD-Untargeted	55.21	55.20	32.89	32.89	65.07	65.07
	1-RR	1000-RR	1-RR	1000-RR	1-RR	1000-RR
PGD 50-step, r-RR	55.30	54.55	32.98	32.09	65.20	65.02

- We also note from Table-9 that the drop in accuracy with 1000 random restarts is marginal.
- We note from Fig.3 that an increase in perturbation bound increases the effectiveness of PGD 7-step attacks, and is able to bring down the accuracy of the model to 0 at large bounds. Further, the loss on FGSM samples monotonically increases in the vicinity of the data samples. These trends indicate the absence of gradient masking.
- We present results against AutoAttack (Croce & Hein, 2020) in Table-1 of the main paper. AutoAttack is an ensemble of several gradient-based attacks and a gradient-free attack Square (Andriushchenko et al., 2020). The robust accuracy against AutoAttack is similar to the accuracy against gradient-based attack GAMA (Sriramanan et al., 2020) indicating that gradient-free attacks are not significantly stronger than gradient based attacks.
- We show the loss surface plots of the proposed defenses ACAT and DAJAT in the vicinity of data samples in Fig.4. We note that the loss surface of the proposed defenses is smooth similar to the TRADES-AWP defense, indicating the absence of gradient masking.

We finally compare the robust accuracy against various attacks in Tables-8 and 9 with the robust accuracy against GAMA attack (Sriramanan et al., 2020) and AutoAttack (Croce & Hein, 2020) in Table-1 of the main paper. The latter evaluations are significantly stronger, indicating that the evaluation presented in the main paper is robust.



D 1RUPDO 7UDEQZQS'(6 $:3 F $&$7 Ours G '$-$7 Ours

Figure 4. **Loss Surface Plots:** Plot of cross-entropy loss in the local neighborhood of images along the gradient direction (δ_1) and a random direction perpendicular to the gradient (δ_2). The loss surface of the proposed defenses ACAT and DAJAT are smooth similar to the TRADES-AWP defense, indicating the absence of gradient masking.