Data-Centric Techniques To Robust ML Models

Jian Gao
Guangxi University

Jingying Tang
Guangxi University

Fengling Lin
Guangxi University

Abstract

With the vigorous technological development, Artificial Intelligence has gradually become the driver of many practical applications. We participated a Data-Centric AI Competition which changes the traditional format and aims to improve a dataset given a fixed model. We convert some online data augmentation methods into offline and generate adversarial examples by the data-centric idea on model robustness. We discover ways to enhance model robustness. In our work, we find that using original data, original training data with style transfer methods and some data augmentations and PGD adversaries will be robustest when the generalization is maximum. Inspired by Stylized-ImageNet, we generate some cartoonization images and it shows that cartoon images can make the model rely less on texture but the shape of the object, which will enhance the robustness of the model. Additionally, RandAugment can enhance the generalization ability, but it will reduce the robustness.

1. Introduction

Deep learning has taken an important place in machine learning with its outstanding performance and it achieves certain functions by fitting massive data. With the emergence of some application scenarios such as face recognition, traffic recognition, and action recognition, the security problems faced by AI have gradually surfaced [1]. For example, in the recognition problem, artificially creating some subtle disturbances to the picture can produce a completely different classification of the picture by a similar model.

The human visual system is more robust in ways than computer vision systems [2]. Deep learning classifiers are deceived by small changes in query images but the human vision system is not and humans are not confused by corruption such as snow, blur, pixelation, and some of or all of these combinations. Humans can even deal with abstract changes in structure and style. Achieving these kinds of robustness is an important goal for computer vision and machine learning.

Though data is significant and large amounts of data drive to the success of AI, it sometimes gets sidelined in the life cycle of ML projects. Current machine learning competitions mostly seek for a high-performance model given a fixed dataset, while recent Data-Centric AI Competition [1], changes the traditional format and aims to improve a dataset given a fixed model. The central objective of data-centric machine learning is working on data.

Similarly, in the aspect of robust learning, many defensive methods have been proposed of deep learning models for mitigating the potential threat of adversarial examples, but most of them strive for a high-performance model in fixed constraints and dataset. Thus how to construct a dataset that is universal and effective for the training of robust models has not been extensively explored.

Our contribution in this paper is:

- In the form of experiments, we study the effectiveness of converting some online data augmentation methods into offline and the adversarial examples generated by the data-centric idea on model robustness.

2. Related Work

Corruptions. Dan Hendrycks and Thomas G. Dietterich [2] introduce common visual corruptions and apply them to the ImageNet object recognition challenge to create ImageNet-C. Their corruption robustness benchmark consists of 15 diverse corruption types which can be divided to noise, blur, weather and digital. These 15 corruption types each have five different levels of severity, since corruptions can manifest themselves at varying intensities.

Style transfer. Convolutional Neural Networks (CNNs) are commonly thought to recognise objects by learning increasingly complex representations of object shapes. Some recent studies suggest a more important role of image textures. Robert Geirhos et al. [3] provide evidence that machine recognition today overly relies on object textures rather than global object shapes as commonly assumed. In order to reduce the texture bias of CNNs, they introduced Stylized-ImageNet (SIN), a data set that removes local cues through style transfer and thereby forces networks to go beyond texture recognition.

Adversarial Examples. Adversarial examples are commonly viewed as a threat to ConvNets. An adversarial image is a clean image perturbed by a small corruption so as to confuse a classifier. Cihang Xie et al. [4] present an opposite perspective: adversarial examples can be used to improve image recognition models if harnessed in the right manner. Key to their method is the usage of a separate auxiliary batch norm for adversarial examples.

PGD. Aleksander Madry et al. [5] stated that the underlying optimization problem is tractable. They provide strong evidence that first-order methods can reliably solve this problem. They supplement these insights with ideas from real analysis to further motivate projected gradient descent (PGD) as a universal "first-order adversary", i.e., the strongest attack utilizing the local first order information about the network. Furthermore, their findings provide evidence that deep neural networks can be made resistant to adversarial attacks.

In our work, we use PGD adversary to generate adversarial examples which improve the robustness of the model.

Lable-Smoothing. Soft targets are a weighted average of the hard targets and the uniform distribution over labels and can improve the generalization and learning speed of a multi-class neural network. Smoothing the labels in this way prevents the network from becoming over-confident. Label-smoothing has been used successfully to improve the accuracy of deep learning models across a range of tasks, including image classification, speech recognition, and machine translation. Rafael Müller et al. [6] demonstrate that label-smoothing implicitly calibrates learned models but impairs distillation, i.e., when teacher models are trained with label smoothing, student models perform worse.

RandAugment. Data augmentation has played a central role in the training of deep vision models. On CIFAR-10, the default augmentations for all methods include flips, pad-and-crop and Cutout. N and M were selected based on the validation performance on 5k held out examples from the training set for 1 and 5 settings for N and M. On the held out 5k dataset, sampled 2 and 4 settings for N and M, respectively (i.e. N = $\{1, 2\}$ and M = $\{2, 6, 10, 14\}$). Results indicate that RandAugment achieves either competitive (i.e. within 0.1%) or state-of-the-art on CIFAR-10 across four network architectures [7].

In this paper, we use the parameters setting N = 1, M = 2 and N = 2, M = 14 and find that RandAugment can enhance the generalization ability.

3. Methology

3.1. Corruptions

We performed corruption methods in [2] on ImageNet-C and applied additional corruptions such as color-change, blur, noise, elastic, weather to images, exemplified in TABLE 1. During data augmentation process, we chose at most two different corruptions at random on the original images, and use augmented images to train the model, which can effectively improve the robustness.

3.2. Cartoonization

With the inspiration of Stylized-ImageNet [3], the shape of the object is protruded through cartoonization, thereby forces the model to rely less on texture recognition but focuses more on the shape, which will achieve greater robustness.

TABLE 1.

ColorChange	Noise	Blur	Elastic	Weather
Brightness	GaussianNoise	DefocusBlur	Superpixels	Snowflakes
Clahe	ImpulseNoise	GaussianBlur	JpegCompression	Snow
Contrast	LaplaceNoise	GlassBlur	Pixelate	Frost
Grayscle	PoissonNoise	MediaBlur	Superpixels	Rain
Saturation	SaltAndPepper	MotionBlur	ElasticTransfor	Fog
	ShotNoise	Spatter		Clouds
	SpeckleNoise	ZoomBlur		FastSnowyLandscape

3.3. Adversarial Examples

Using the PGD attack [5] method to generate adversarial examples for adversarial training can improve the robustness of the model. In the competition, we use the highest score of non-adversarial example model as attack target and generate PGD adversaries as the ultimate data augmentation. As emphasized in [4], we use the adversarial examples with enhanced original samples as the training input data for training model and use KLDivLoss which can improve the model robustness as well.

3.4. Label-Smoothing

We mainly use label-smoothing method [6]. This method can improve the generalization ability of the model, but it will slightly reduce the robustness. The original data are 10k images in CIFAR-10 validation dataset.

3.5. RandAugment

We use RandAugment [7] for the CIFAR-10 dataset, we try N = 1, M = 2 and N = 2, M = 14 and find that the effect is similar to label-smoothing. Additionally, RandAugment can improve the generalization ability of the model, but it will reduce the robustness.

4. Experiment

This competition consists of two stages. In Stage I, organizer choose 2 baseline networks on CIFAR-10. These models come from: ResNet50 [8] and DenseNet121 [9]. In Stage II, models come from PreActResNet18 [10] and WideResNet [11]. 60k images in CIFAR-10 need to be generated into no more than 50k images. These two stages both use Kullback-Leibler divergence Loss (KLDivLoss) as loss function. According to the rules: models and loss function are specified, epoch $\leq$ 200, batchsize $\leq$ 256. In Stage II, we use the robustness test set (confidential to contestants) provided by the competition to do the online verification. The score formula is shown in in Eqn.(1) where $\mathcal{M}$ is the set of all trained models, $\mathcal{X}$ is the evaluation

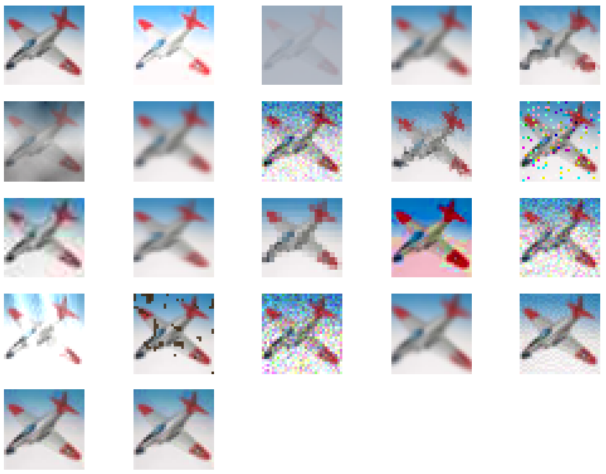

Figure 1. 1th images: is an original image. 2nd to 19th images: are examples of corruptions augmentation. Remaining iamges: are three kinds of unknown adversarial examples.

dataset. Whenever multiple submissions obtain the same score, they will be compared by the number of data points (less is better). This is a completely data-centric robustness research competition.

$$Score = \frac{1}{|\mathcal{M}|} \sum_{\mathcal{M}_i \in \mathcal{M}} \frac{1}{|\mathcal{X}|} \sum_{(x_j, y_j) \in \mathcal{X}} 1(\mathcal{M}_i(x_i) = y_i) \quad (1)$$

4.1. Dataset

After the competition, the test dataset was released. Stage I directly uses 10k original *val* datatset on CIFAR-10 as the basic data. In Stage II these images are randomly selected. The data distribution of these two datasets are the same which consist of original data, original data with corruption augmentation and adversarial examples of original data. In Stage I, corruption augmentation uses noise, blur and weather. In Stage II corruption augmentation adds color-change and elastic on the basis of Stage I. The total data is the original image data×13 and in Stage II is the original image data×22. The compesition of these two datasets are showed in TABLE 2, 3, and image examples appear in Figure 1.

In our experiment, we call the number of 10k *val* dataset on CIFAR-10 as 'coefficient' which equals to 10k. We superimpose all the methology above: the original data with coefficient×1, corruption coefficient×2, cartoon coefficient×1 and PGD coefficient×1. We generated new datasets to improve the robustness in two stages but we used label-smoothing only in Stage II . Example images in the new dataset are showed in Figure 2.

4.2. Training parameters

The competition we participated is AAAI-2022 Security AI Challenger Program Phase 8: The data-centric robust machine learning. The *train* file is provided by the competition,

TABLE 2. STAGE I DATASET.

Original data	10k
Adversarial examples	10k×2
	Unknown adversarial examples1×10k
	Unknown adversarial examples2×10k
Corruptions	10k×10
	ShotNoise×10k
	ImpulseNoise×10k
	GaussianNoise×10k
	ZoomBlur×10k
	GlassBlur×10k
	DefocusBlur×10k
	MotionBlur×10k
	Snow×10k
	Frost×10k
	Fog×10k

Figure 2. *Top row:* The first one is an original image in CIFAR-10. The second one is an PGD adversarial example. The third one is an example of stylized transfer. *Remaining:* augmentated images by randomly selecting 2 corruption strategies.

the ResNet50 and DenseNet121 models are used in Stage I, and the WideresNet and PreactresNet18 models are used in Stage II.

The highest score model training parameters: In ResNet50 we use SGD as optimizer, learning_rate$_{start}$ = 0.01, momentum = 0.9, weight_decay = 0.0005, scheduler is CosineAnnealingWarmRestarts, epochs = 185, the number of iterations for the first restart is 3, set an increasing factor of 2, learning_rate$_{min}$ = 0.00001, minibatch_size = 128. In DenseNet121 we use SGD as optimizer, learning_rate$_{start}$ = 0.1, momentum = 0.9, weight decay = 0.0001, scheduler is CosineAnnealingLR, epochs = 200, learning_rate$_{min}$ = 0.001, minibatch_size = 128. In WideResNet model we use SGD as optimizer, learning_rate$_{start}$ = 0.1, momentum= 0.9, weight_decay = 0.0001, scheduler is CosineAnnealing-

TABLE 3. STAGE II DATASET.

Original data	10k
Adversarial examples	10k×3
	Unknown adversarial examples1×$10k$
	Unknown adversarial examples2×$10k$
	Unknown adversarial examples3×$10k$
Corruptions	10k×18
	Brightness×$10k$
	Contrast×$10k$
	Saturate×$10k$
	ShotNoise×$10k$
	ImpulseNoise×$10k$
	GaussianNoise×$10k$
	SpeckleNoise×$10k$
	ZoomBlur×$10k$
	GlassBlur×$10k$
	DefocusBlur×$10k$
	MotionBlur×$10k$
	Spatter×$10k$
	Snow×$10k$
	Fog×$10k$
	Elastictransform×$10k$
	Jpegcompression×$10k$
	Pixelate×$10k$

WarmRestarts, trained for 200 epochs, learning_rate$_{min}$ = 0.0001, minibatch_size = 128. In PreactResNet18 model we use SGD as optimizer, learning_rate$_{start}$ = 0.05, momentum= 0.9, weight_decay = 0.0005, scheduler is CosineAnnealingWarmRestarts, epochs = 200, learning_rate$_{min}$ = 0.0005, minibatch_size = 128.

4.3. Impact of Data Combination on Robustness

TABLE 4. TRAINING PARAMETERS.

	Schedule	Batchsize	Model
I	CosineAnnealingLR 200	128	ResNet50
	CosineAnnealingLR 200	128	DenseNet121
II	CosineAnnealingWarmRestarts 185	256	ResNet50
	CosineAnnealingLR 200	256	DenseNet121
III	CosineAnnealingWarmRestarts 185	128	ResNet50
	CosineAnnealingLR 200	128	DenseNet121
IV	CosineAnnealingWarmRestarts 185	256	WideResNet
	CosineAnnealingLR 200	256	PreActResNet18
V	CosineAnnealingLR 200	128	WideResNet
	CosineAnnealingLR 200	256	PreActResNet18

Using cartoon augmentation alone is better than using with other augmentation techniques. We use cartoon augmentation in the serie of data corruption augmentation, but the performance is not good as it used alone. Corruptions affect the cartoon images and make the model pay more attention to the properties of the contour. This experiment use I parameters in TABLE 4. Results appear in TABLE 5.

TABLE 5.

Scenario	Stage I - Score
Baseline	74.12
ORI×1, PGD×1, Corruption×3	98.23
ORI×1, PGD×1, Cartoon×2, Corruption×1	98.08
ORI×1, PGD×1, Cartoon×1, Corruption×2	98.44

The image data generated by MixUp, Cut-Mix, RICAP are not good as the original integral images, these methods require more data caps and training steps. In epoch$\leq$ 200 and data_size $\leq$ original_data_size $\times$5: MixUp $>$ Cut-Mix $>$ RICAP . Results are showed in TABLE 6.

TABLE 6.

Scenario	Stage I - Score
Baseline	74.12
ORI×1, Corruption×1, CutMix-Corruption×3	95.90
ORI×1, Corruption×1, RICAP-Corruption×3	96.59
ORI×1, Corruption×1, Mixup-Corruption×3	97.37
ORI×1, Corruption×4	97.51

Among adversarial examples, we try adversarial attacks such as FGSM [12], I-FGSM [13], CW-L2 [14], Sparse-L1-descent [15], SPSA [16] and PGD. We find that PGD adversaries perform better on robustness than others and it is better used with with original data. Results appear in TABLE 7.

TABLE 7.

Scenario	Param(Tab.4)	Stage I - Score
Baseline		74.12
ORI×1, FGSM×1, Cartoon×1, Corruption×2	I	97.86
ORI×1, 6-Attacks×1, Cartoon×1, Corruption×2	I	98.10
I-FGSM×1, PGD×1, Cartoon×1, Corruption×2	II	98.47
CW-L2×1, PGD×1, Cartoon×1, Corruption×2	II	98.59
ORI×1, PGD×1, Cartoon×1, Corruption×2	II	98.59
ORI×1, PGD×1, Cartoon×1, Corruption×2	III	98.75

Adversarial examples are better not be used with other data augmentation methods, we use our data corruption augmentation to the adversarial examples, and performed a worse score, adversarial examples are the ultimate data augmentation method. This experiment use II parameters in TABLE 4. Experiment results are showed in TABLE 8.

TABLE 8.

Scenario	Stage I - Score
Baseline	74.12
CW-L2×1, PGD×1, Cartoon×1, Corruption×2	98.59
CW-L2-Corruption×1, PGD-Corruption×1, Cartoon×1, Corruption×2	96.81

Val-10K	Schedule	Imgaug(2 of 5)	Cartoon	Adverisal	Albu	Mix-up	Cut-mix	RICAP	RA-2-14	KIP/LS	Score
√											74
√	cos200X2				√	√	√	√			89.04
	cos200X2	√			√	√	√	√			94.42
	cos200X2	√	√		√	√	√	√	√		95.82
√	cos200X2	√	√			√	√	√	√		96.46
√	cos200X2	√	√				√				95.90
√	cos200X2	√	√					√			96.59
√	cos200X2	√	√			√					97.37
√	cos200X2	√	√		O-S						97.51
√	cos200X2	√	√		O-S				√		96.81
√	cos200X2	√	√		O-S					KIP-10K	96.25
√	cos200X2	√	√	FGSM	O-S						97.79
√	cos200X2	√	√	PGD	O-S						98.44
√	cos200X2	√	√	6-attacks	O-S						98.10
√	cos200X2	√	√	PGD	O-S					KIP-500	98.39
√	cos200X2	√	√	PGD	O-S					LS	98.33
√	warm-up185cos200	√	√	PGD	O-S						98.58
	warm-up185cos200	√	√	PGD,CW-L2	O-S						98.59
	warm-up185cos200	√	√	PGD,I-FGSM	O-S						98.46
√	warm-up185cos200 B: 128	√	√	PGD	O-S						98.75
6-attacks=(PGD,FGSM,I-FGSM,CW-L2,SPSA,Sparse-L1-descent), B:128=batch_size: 128, O-S=only use image shift.											

Figure 3. Results in Stage I. All the changes are based on 10k images in CIFAR-10 validation dataset.

Label-smoothing can improve generalization of robust model, we use the train dataset in CIFAR-10 as generalization ability validation dataset. We find that the label-smoothing can increase the generalization ability, but slightly decrease the robustness. For the test dataset was changed in Stage II, the model need stronger generalization ability. In Stage II, we use label-smoothing and RandAugment (N = 2, M = 14) for 10k original data. This experiment use I parameters in TABLE 4. Results appear in TABLE 9.

TABLE 9.

Scenario	Validation	Stage I - Score
Baseline	-	74.12
ORI×1, PGD×1, Cartoon×1, Corruption×2	73.45	98.44
(ORI×1, PGD×1, Cartoon×1, Corruption×2) LS	74.49	98.33

The test dataset in Stage I is 10K images of the *val* dataset in CIFAR-10, we use the parameter N = 2, M = 14 in [7] to replace the corruption×1 in our work. We use the *train* dataset in CIFAR-10 as generalization ability validation dataset. We observe that RandAugment can significantly improve the generalization ability, but it will cause a small loss in robustness. This experiment use I parameters in TABLE 4. Results appear in TABLE 10.

TABLE 10.

Scenario	Validation	Stage I - Score
Baseline	-	74.12
ORI×1, Corruption×4	70.60	97.51
ORI×1, RA-2-14×1, Corruption×3	80.38	96.81

In our experiment, we choose the *val* dataset in CIFAR-10 as the basic data to improve the robustness in Stage I. In Stage II, we choose the *train* dataset in CIFAR-10 and divide it into 5 equal parts as the basic data to improve the generalization ability and robustness. Results are showed in TABLE 11.

We illustrate our results in Stage I in Figure 3.

TABLE 11.

Scenario		Stage I - Score
Baseline		74.12
ORI×1, PGD×1, Cartoon×1, Corruption×2		98.75
Scenario (all labels use label-smoothing)	Param (Tab.4)	Stage II - Score
Baseline	-	63.31
ORI×1, PGD×1, Cartoon×1, Corruption ×2 [1]	IV	76.01
Corruption ×5 [2]	IV	78.41
PGD×2, Cartoon×1, Corruption ×2 [2]	V	83.75
ORI×1, PGD×1, Cartoon×1, Corruption ×2 [2]	V	85.40
ORI-RA-2-14×1, PGD×1, Cartoon×1, Corruption×2 [2]	V	85.43

[1] Train on *val* dataset in CIFAR-10.
[2] Train on *train* dataset in CIFAR-10.

5. Conclusion and Future Work

After all the experiments above, we use the original images adding a slight shift of 40% chance. Cooperate with cartoon and some 2 of 5 data augmentation strategies to increase the interference intensity and the diversity of data augments. After obtaining a certain strength of data, the adversarial attack experiment was carried out on the model trained with these data, and the optimal attack method PGD was determined to generate the adversarial samples. Finally, adversarial training combined with the previous data-enhanced samples achieved a score of 98.75 in the Tournament Stage I test dataset and a score of 83.75 in the Tournament Stage II test dataset. Cartoon augmentation are not suitable for simultaneously using with other augmentation techniques. The image data generated by MixUp, CutOut, Cut-Mix, RICAP and etc. are not good as the original integral images. PGD performed better in robustness. Adversarial examples are ultimate. Label-smoothing can improve generalization.

The data upper cap of this competition is numerical not a proportional base, we would like to do the research on combination strategies of data augmentation with data growing

linearly in base rate. Since the limitation of submissions, we have no chance to test the basic model on generating adversarial examples, we will test the relationship between the basic model, train dataset, methods and convergence. In corruption augmentation, all the methods are randomly selected which make the score (training performance) fluctuate. We attempt to train solidified methods such as AutoAugment [17], RandAugment and etc. on our generated dataset to replace completely random selection. Since our own validation dataset can only verify the generalization ability or robustness of individual model, it is hoped that a general validation dataset or a generation method is suitable for both model generalization ability and robustness so that data-centric research can be more directly and effectively. We hope to try methods in the direction of data distillation [18] to expand data-centric research in the future.

Acknowledgments

We would like to thank the security AI challenger program launched by Alibaba Group and Tsinghua University and this work gets bonus in the competition.

References

[1] AAAI2022 Security AI Challenger VIII: Data-Centric Robust Learning on ML Models competition. https://https-deeplearning-ai.github.io/data-centric-comp/.

[2] Dan Hendrycks, Thomas G. Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Surface Variations. arXiv:1807.01697, 2018.

[3] Robert Geirhos, Patricia Rubisch, Claudio Michaelis and Matthias Bethge, Felix A. Wichmann, Wieland Brendel. IMAGENET-TRAINED CNNS ARE BIASED TOWARDS TEXTURE; INCREASING SHAPE BIAS IMPROVES ACCURACY AND ROBUSTNESS. *In ICLR*, 2019.

[4] Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang and Alan Yuille and Quoc V. Le. Adversarial Examples Improve Image Recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.819-828, 2020.

[5] Aleksander M adry, Aleksandar Makelov, Ludwig Schmidt. Towards Deep Learning Models Resistant to Adversarial Attacks. arXiv:1706.06083, 2017.

[6] Rafael Müller, Simon Kornblith, Geoffrey Hinton. When Does Label Smoothing Help? arXiv:1906.02629, 2019.

[7] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, Quoc V. Le. RandAugment: Practical automated data augmentation with a reduced search space. arxiv.org/abs/1909.13719, 2019.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. arxiv.org/abs/1512.03385, 2015.

[9] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. Densely Connected Convolutional Networks. arXiv:1608.06993, 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Identity Mappings in Deep Residual Networks. arxiv.org/abs/1603.05027, 2016.

[11] Sergey Zagoruyko, Nikos Komodakis. Wide Residual Networks. arxiv.org/abs/1605.07146, 2016.

[12] Ian J. Goodfellow, Jonathon Shlens, Christian Szegedy. Explaining and Harnessing Adversarial Examples. arxiv.org/abs/1412.6572, 2014.

[13] Alexey Kurakin, Ian J. Goodfellow, Samy Bengio. ADVERSARIAL EXAMPLES IN THE PHYSICAL WORLD. arxiv.org/pdf/1607.02533, 2017.

[14] Nicholas Carlini, David Wagner. Towards Evaluating the Robustness of Neural Networks. arxiv.org/abs/1608.04644, 2016.

[15] Florian Tramèr, Dan Boneh. Adversarial Training and Robustness for Multiple Perturbations. arxiv.org/pdf/1904.13000, 2019.

[16] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, Pushmeet Kohli. Adversarial Risk and the Dangers of Evaluating Against Weak Attacks. arxiv.org/abs/1802.05666, 2018.

[17] Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, Quoc V. Le. AutoAugment: Learning Augmentation Policies from Data. arXiv:1805.09501, 2019.

[18] Timothy Nguyen, Roman Novak, Lechao Xiao, Jaehoon Lee. Dataset Distillation with Infinitely Wide Convolutional Networks. *NeurIPS*, 2021.