SCALING COMPUTE IS NOT ALL YOU NEED FOR ADVERSARIAL ROBUSTNESS

Edoardo Debenedetti
ETH Zurich

Zishen Wan
Georgia Institute of Technology

Maksym Andriushchenko
EPFL

Vikash Sehwag
Princeton University

Kshitij Bhardwaj
Lawrence Livermore National Lab

Bhavya Kailkhura
Lawrence Livermore National Lab

ABSTRACT

The last six years have witnessed significant progress in adversarially robust deep learning. As evidenced by the CIFAR-10 dataset category in RobustBench benchmark, the accuracy under ℓ_∞ adversarial perturbations improved from 44% in Madry et al. (2018) to 71% in Peng et al. (2023). Although impressive, existing state-of-the-art is still far from satisfactory. It is further observed that best-performing models are often very large models adversarially trained by industrial labs with significant computational budgets. We aim to understand: "how much longer can computing power drive adversarial robustness advances?" To answer this question, we derive *scaling laws for adversarial robustness* which can be extrapolated in the future to provide an estimate of how much cost we would need to pay to reach a desired level of robustness. We show that increasing the FLOPs needed for adversarial training does not bring as much advantage as it does for standard training in terms of performance improvements. Moreover, we find that some top-performing techniques are difficult to exactly reproduce, suggesting that they are not robust enough for minor changes in the training setup. Our analysis also uncovers potentially worthwhile directions to pursue in future research.

1 INTRODUCTION

Rapid advances in deep learning continue to improve performance on benchmarks across data modalities, achieving near-human, and in some cases even better than human performance. However, the challenge of achieving very high adversarial robustness remains elusive, despite substantial investmentment in robust learning mechanisms. While scaling of computational resources has become a de facto approach to drastically improve the performance of deep neural networks across numerous tasks (Bahri et al., 2021; Bansal et al., 2022; Gordon et al., 2021; Kaplan et al., 2020; Krishnan et al., 2022; Sharma and Kaplan, 2022; Zhai et al., 2022), similar studies are nearly missing in the context of adversarial robustness. In this work, we take some first steps toward answering the question: *Is scaling compute a viable solution for achieving adversarial robustness?* Note that scaling of computational resources in deep learning has a direct increase in environmental impact. Thus we are further interested in a two-fold impact of scaling, i.e., whether continuing a "scaling compute" strategy would provide proportional improvement in adversarial robustness or would it have a disproportional environmental impact without much benefit to adversarial robustness.

To achieve this objective, we carry out a systematic empirical exploration by adversarially training a large number of models on CIFAR-10 dataset, the most commonly studied dataset on this task. Specifically, we vary several factors such as model size, adversarial loss, attack steps, use of synthetic data, etc. Next, we analyze the adversarial robustness of the resulting adversarially trained model zoo. Contrary to the intuitive notion that greater computational power (i.e., deeper and more complex models) should inherently improve a model's ability to withstand adversarial attacks, our research reveals an interesting phenomenon: *scaling up model sizes does not yield proportionate improvements in adversarial robustness*—setting adversarial training apart from standard training. This implies that exponentially more computational resources (and carbon emission) are required to achieve similar gains, making compute scaling an inefficient (if not infeasible) approach to solving this open problem.

We argue that the adversarial robustness community must think beyond the traditional paradigm of simply increasing compute. We must explore innovative avenues for addressing this important challenge, one that has clearly defied conventional approaches and has profound implications for the security and reliability of modern machine learning systems. In this work, we highlight the limitations of the current strategies and advocate for more creative and holistic approaches to achieving adversarial robustness, urging the field to break new ground in pursuit of robust AI systems.

2 Experimental Roadmap

We run extensive experiments over several hyper-parameters, spanning design choices such as loss function, architecture, activation function, number of attack steps, and amount of data used. *Overall, we train 320 models*, requiring more than ten thousand GPU hours of training.

Robust Losses The first axis we ablate is the adversarial training loss: we compare the original Adversarial Training (AT) loss (introduced earlier in Section 4.1) to the TRADES (Zhang et al., 2019) one. Briefly speaking, TRADES reformulates the original PGD loss as a regularization term to improve the trade-off between robust and standard accuracy. In order to do this, it has two loss components, weighted by a parameter β used to balance the trade-off. The two components are the loss on the original input $\boldsymbol{x}$, and the loss on the adversarial example found by maximizing the Kullback–Leibler divergence between the model output given the original input $\boldsymbol{x}$ and given the adversarial example $\boldsymbol{x} + \boldsymbol{\delta}$. Formally, the loss is:

$$\min_{f} \mathbb{E}_{(\boldsymbol{x},y)\sim D}\big[\ell(f(\boldsymbol{x}), y) + \beta \max_{\delta\in\Delta} \ell_{KL}(f(\boldsymbol{x}), f(\boldsymbol{x}+\boldsymbol{\delta}))\big]. \tag{1}$$

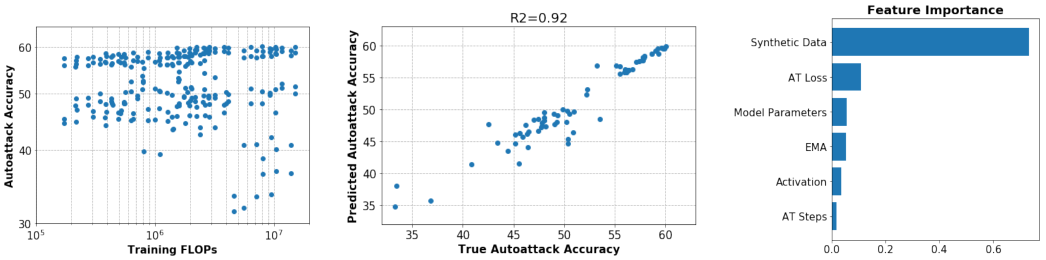
Interestingly, the TRADES and the AT losses require different amounts of compute: for TRADES, we need to run one forward pass in addition to the number of forward and backward passes needed to compute the adversarial example for the given training input, while for the PGD loss, we need to run only as many forward passes as the number of attack iterations. However, we note that the additional forward pass required by TRADES gets amortized when the number n of attack steps is larger. For example, for WideResNet-28-10 with 1 attack steps, TRADES increases the number of FLOPs for one epoch by $1.3\times$, while it increases by only $1.08\times$ when using 10 iterations.

Models We test several sizes of WideResNet (Zagoruyko and Komodakis, 2016), with different depths and widths, i.e., 28-10, 34-10, 34-20, and 70-16. This is a common architecture in the adversarial training literature, and these are the most used size configurations for this architecture (Croce et al., 2020). Studying these architecture sizes enables us to analyze models with a wide range of FLOPs per forward pass and parameters, from 10.5 GFLOPs and $36M$ parameters for WideResNet-28-10 to 77.6 GFLOPs and $266M$ parameters for WideResNet-70-16. Moreover, we also test two different activation functions: ReLU, used originally in WideResNet, and GELU (Hendrycks and Gimpel, 2016). Previous work observed that smooth activation functions improve robustness (Xie et al., 2020b; Debenedetti et al., 2023). However, also GELU comes at extra compute cost as it requires about $1.5\times$ the memory required by ReLU.

Attack Steps We vary the number of attack iterations, training our models with 1, 2, 5, 7, and 10 iterations. Increasing attack iterations leads to stronger attacks. Obviously, training models with more steps require more FLOPs as each attack iteration requires one additional forward and one additional backward pass over the model.

Number of Training Epochs and Exponential Moving Average We follow both the original training set-up from Zhang et al. (2019), with 100 training epochs and no exponential moving average (EMA) of the model weights, and the set-up proposed by Gowal et al. (2020). The latter comprises training the model for 400 epochs with EMA of the model weights with a decay of 0.995 . This setup, not only requires $4\times$ the number of FLOPs for the overall training, but it also requires $2\times$ the memory as we need to keep both the weights being trained and those being averaged through EMA.

Synthetic Data Finally, we test the contribution achieved by synthetic data for adversarial training. Previous work shows that extra unlabeled data provides a significant boost in performance (Carmon



(a) Training FLOPs vs Autoattack accuracy in log-log space.

(b) Performance of GBR regressor on autoattack accuracy prediction task.

(c) Feature importance of GBR model.

Figure 1: Analysis of various algorithmic changes on adversarial robustness.

et al., 2019) and that this holds also if the extra data is synthetically generated (Gowal et al., 2021; Sehwag et al., 2021; Wang et al., 2023). For this reason, we also test how much synthetic data helps with robustness in all the aforementioned scenarios. We use 1M synthetic CIFAR-10 images released by Gowal et al. (2021). However, we do not account for the synthetic data generation nor the generative model training in our FLOP count. This is mainly because there are many open-source pre-trained models that can be used for generating data, and, overall, the compute needed to generate the data is negligible compared to the overall training compute, and the synthetic data can be used across different training runs, hence the compute amortized.

The detailed accuracy and efficiency evaluation metrics are elaborated in Appendix Section 5.

3 Results

In this section, we analyze the data generated in Section 2 to answer our questions of interest. As can be seen from Figure 1a, FLOP counts of our training runs span two orders of magnitude, and the generated data is sufficiently diverse.

3.1 What Algorithmic Choices are Driving Adversarial Robustness Advances?

In this section, we aim to unravel which algorithmic interventions are important for advancing adversarial robustness. We first try to predict how well a model (using a specific adversarial training recipe) can perform without actually training or testing it. Specifically, we use number of model parameters, synthetic data $\in$ {True, False}, activation $\in$ {ReLU, GELU}, loss $\in$ {PGD, TRADES}, PGD steps $\in$ {1, 2, 5, 7, 10}, and exponential moving average (EMA) $\in$ {True, False} as input features to our predictive model. We use gradient boosting regression model with number of estimators equal to 50 and max depth equal to 5. We use a 70:30 train-test split in our experiment.

In Figure 1b, we plot AutoAttack accuracy for unseen/holdout adversarial training configurations predicted by our model against ground truth values. We see that our model can predict the performance of unseen test configurations with high accuracy.

To the best of our knowledge, this is the first attempt to show that the adversarial robustness of a given training recipe can directly be predicted without going through a computationally expensive training-validation cycle.

This finding is expected to have broad implications for compute-efficient training (Bartoldson et al., 2023) and reliable testing (Zhang et al., 2020) of machine learning models.

Next, knowing which algorithmic knobs are important for predicting AutoAttack accuracy can be very useful in helping designers understand the problem and devising better approaches for improving the adversarial robustness. Thus, we analyze the feature importance of our predictive model in Figure 1c. Feature importance provides a score that indicates how useful (or valuable) each feature is for our predictive model. These importance scores are calculated individually for each feature in our dataset, enabling the features to be ranked and compared to each other in Figure 1c.

We hypothesize that the use of synthetic data, training loss, and the number of model parameters are the most important driving forces behind the adversarial robustness advances.

We take a closer look into this finding in Figure 2. We see that in the data-limited regime, i.e., without synthetic data, the choice of the loss function (PGD vs. TRADES) or regularization (EMA vs No EMA) plays an important role in improving the adversarial robustness of a given model – evidenced by high variance in autoattack accuracy. However, in the presence of synthetic data, the variance due to these choices is reduced, and model size plays a greater role. These findings are reasonable and also evident from popular leaderboards such as RobustBench (Croce et al., 2020).

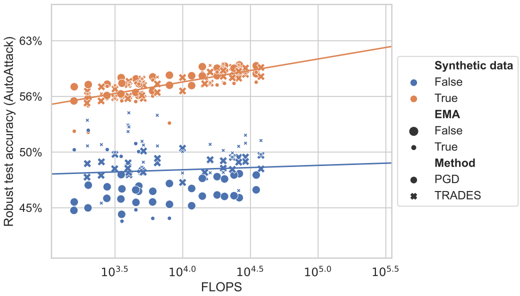


Figure 2: A deeper look into the effect of various algorithmic changes on adversarial robustness.

3.2 What is the Computational Burden of Adversarial Robustness?

Having identified promising strategies for improving adversarial robustness in Section 3.1, in this section, we explore the limits of scaling compute (i.e., FLOPs) to achieve adversarial robustness. Specifically, we study empirical scaling laws for adversarial robustness performance in terms of AutoAttack accuracy. We first divide the FLOP range into 19 equisized bins. Then, for each FLOP bin, we determine which run achieves the best AutoAttack accuracy. This serves as the best performance envelope for our FLOP range in consideration. Similarly to Thompson et al. (2020), we expect a polynomial relationship between compute and AutoAttack accuracy:

$$\text{accuracy} = C \times (\text{FLOPs})^{\alpha}, \tag{2}$$

where C and α are learnable parameters. Next, we fit power laws to estimate the best AutoAttack accuracy for any given amount of compute.

Figure 3a in Appendix shows the growth in the AutoAttack accuracy as a function of the computation used in these models. Given that this is plotted on a log-log scale, a straight line indicates a polynomial growth in computing per unit of performance. In other words, there is a clear power law relationship between the adversarial robustness and compute spanning two orders of magnitude (with $R2 = 0.76$). We find that the power law relationship also holds for electricity cost (see Figure 3b) and CO2 emission (see Figure 3c) both with $R2 = 0.95$.

AutoAttack accuracy, electricity cost, and CO2 emission of adversarial training follow a power law with respect to compute (i.e., FLOPs).

Unfortunately, the growth rate for AutoAttack accuracy is very small as seen from the estimated parameters $\alpha^* = 0.01$ and $C^* = 50.86$. On the other hand, the growth rate for CO2 emission (electricity cost) is quite large, i.e., $\alpha^* = 0.95$ with $C^* = 7.0 \cdot 10^{-6}$. These findings suggest that scaling compute alone may not be able to help solve the adversarial robustness problem.

Slow growth rate (w.r.t. compute) for adversarial robustness but high growth rates for CO2 emission and electricity cost imply that scaling compute is neither effective nor efficient.

Conversely, it has been observed that, for standard training, much higher growth rates hold hence suggesting that scaling compute is much more effective than it is for adversarial training.

Finally, we note, from Figure 2, that *the usage of extra data (despite being synthetic) has significantly more advantageous scaling laws*, compared to the models trained without extra data.

Additional experimental results and our outlooks regarding the evaluated adversarial training techniques are further discussed in Appendix Section 6 and Section 7.

References

Ibrahim M Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision. *Advances in Neural Information Processing Systems*, 35:22300–22312, 2022.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *arXiv preprint arXiv:2102.06701*, 2021.

Yamini Bansal, Behrooz Ghorbani, Ankush Garg, Biao Zhang, Colin Cherry, Behnam Neyshabur, and Orhan Firat. Data scaling laws in nmt: The effect of noise and architecture. In *International Conference on Machine Learning*, pages 1466–1482. PMLR, 2022.

Brian R Bartoldson, Bhavya Kailkhura, and Davis Blalock. Compute-efficient deep learning: Algorithmic trends and opportunities. *Journal of Machine Learning Research*, 24:1–77, 2023.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. Unlabeled data improves adversarial robustness. *NeurIPS*, 2019.

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

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Edoardo Debenedetti, Vikash Sehwag, and Prateek Mittal. A light recipe to train robust vision transformers. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 225–253. IEEE, 2023.

Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019.

HS Eggleston, Leandro Buendia, Kyoko Miwa, Todd Ngara, and Kiyoto Tanabe. 2006 ipcc guidelines for national greenhouse gas inventories. *Intergovernmental Panel on Climate Change*, 2006.

Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. URL `https://github.com/MadryLab/robustness`.

Dou Goodman, Hao Xin, Wang Yang, Wu Yuesheng, Xiong Junfeng, and Zhang Huan. Advbox: a toolbox to generate adversarial examples that fool neural networks. *arXiv preprint arXiv:2001.05574*, 2020.

Mitchell A Gordon, Kevin Duh, and Jared Kaplan. Data and parameter scaling laws for neural machine translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5915–5922, 2021.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv*, 2020.

Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34, 2021.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43, 2020.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Marius Hobbhahn Jsevillamol. What's the backward-forward FLOP ratio for Neural Networks?, 2021. URL `https://www.lesswrong.com/posts/fnjKpBoWJXcSDwhZk/what-s-the-backward-forward-flop-ratio-for-neural-networks`. Accessed: 2023-10-12.

Daniel Kang, Yi Sun, Tom Brown, Dan Hendrycks, and Jacob Steinhardt. Transfer of adversarial robustness between perturbation types. *arXiv preprint arXiv:1905.01034*, 2019.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Srivatsan Krishnan, Max Lam, Sharad Chitlangia, Zishen Wan, Gabriel Barth-maron, Aleksandra Faust, and Vijay Janapa Reddi. QuaRL: Quantization for fast and environmentally sustainable reinforcement learning. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

Yaxin Li, Wei Jin, Han Xu, and Jiliang Tang. Deeprobust: A pytorch library for adversarial attacks and defenses. *arXiv preprint arXiv:2005.06149*, 2020.

Andrew Lohn and Micah Musser. Ai and compute: How much longer can computing power drive artificial intelligence progress. *Center for Security and Emerging Technology (CSET)*, 2022.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Marco Melis, Ambra Demontis, Maura Pintor, Angelo Sotgiu, and Battista Biggio. secml: A python library for secure and explainable machine learning. *arXiv preprint arXiv:1912.10013*, 2019.

Maria-Irina Nicolae, Mathieu Sinn, Minh Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. Adversarial robustness toolbox v1.2.0. *arXiv preprint arXiv:1807.01069*, 2018.

Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, and Rujun Long. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

ShengYun Peng, Weilin Xu, Cory Cornelius, Matthew Hull, Kevin Li, Rahul Duggal, Mansi Phute, Jason Martin, and Duen Horng Chau. Robust principles: Architectural design principles for adversarially robust cnns. *arXiv preprint arXiv:2308.16258*, 2023.

Rahul Rade. PyTorch implementation of uncovering the limits of adversarial training against norm-bounded adversarial examples, 2021. URL `https://github.com/imrahulr/adversarial_robustness_pytorch`. Accessed: 2023-10-12.

Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *ICML Reliable Machine Learning in the Wild Workshop*, 2017.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *NeurIPS*, 2020.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, 2020.

Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Robust learning meets generative models: Can proxy distributions improve adversarial robustness? In *International Conference on Learning Representations*, 2021.

Utkarsh Sharma and Jared Kaplan. Scaling laws from the data manifold dimension. *The Journal of Machine Learning Research*, 23(1):343–376, 2022.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Dumitru Erhan Joan Bruna, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2013.

Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 2020.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.

Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. *arXiv preprint arXiv:2302.04638*, 2023.

Cihang Xie, Mingxing Tan, Boqing Gong, Jiang Wang, Alan L Yuille, and Quoc V Le. Adversarial examples improve image recognition. In *CVPR*, 2020a.

Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020b.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113, 2022.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

Jie M Zhang, Mark Harman, Lei Ma, and Yang Liu. Machine learning testing: Survey, landscapes and horizons. *IEEE Transactions on Software Engineering*, 48(1):1–36, 2020.

APPENDIX

4 BACKGROUND AND RELATED WORK

4.1 ADVERSARIAL ROBUSTNESS

Background and Threat Model For a model $f : \mathcal{X} \to \mathbb{R}^C$, input $\boldsymbol{x} \in \mathcal{X}$, and label $y \in \{1, \ldots, C\}$, we say that an adversarial perturbation $\boldsymbol{\delta} \in \Delta$ is *successful* if

$$\underset{c \in \{1,\ldots,C\}}{\arg\max}\ f(\boldsymbol{x} + \boldsymbol{\delta})_c \neq y. \tag{3}$$

Following the previous established works (Croce et al., 2020; Madry et al., 2018; Szegedy et al., 2013), we assume that the attacker has *full knowledge* about f (i.e., a white-box threat model), and we measure adversarial robustness via *robust accuracy* for the ℓ_∞ norm, i.e., the fraction of points x on which the model f predicts the correct class y for all possible perturbations δ from $\Delta = \{\boldsymbol{\delta} \in \mathcal{X} \mid \|\boldsymbol{\delta}\|_\infty \leq \varepsilon\}$. We note that robustness to small ℓ_∞-perturbations is of interest in different contexts: for robustness to unseen perturbations (Kang et al., 2019), better gradient-based interpretability (Tsipras et al., 2019), feature learning (Salman et al., 2020), generalization performance (Xie et al., 2020a). Many current robust training approaches are derived from *adversarial training* which is formulated as the following min-max robust optimization problem in Madry et al. (2018):

$$\min_f \mathbb{E}_{(\boldsymbol{x},y)\sim D}\big[\max_{\delta \in \Delta} \ell(f(\boldsymbol{x} + \boldsymbol{\delta}), y)\big], \tag{4}$$

where the inner maximization is typically approximated by multiple iterations of projected gradient descent (PGD). Importantly, since PGD has to be performed on every iteration of training, this significantly increases the computational cost of robust training.

Existing Robustness Benchmarks There are multiple popular robustness benchmarks and libraries in the community. However, importantly, none of these benchmarks take into account the computational costs needed to train state-of-the-art robust models. Most libraries focus primarily on robustness *evaluation* and implement various white- and black-box adversarial attacks (Papernot et al., 2018; Rauber et al., 2017; Nicolae et al., 2018; Ding et al., 2019; Melis et al., 2019; Goodman et al., 2020; Li et al., 2020). Other libraries focus on implementing defenses, most prominently Nicolae et al. (2018) implement multiple defenses from the literature and Engstrom et al. (2019) provides an implementation of adversarial training. As for benchmarks, one of the first benchmarks for different *attacks* is introduced in the works of Madry et al. (2018) and Zhang et al. (2019) whose models withstood many thorough robustness evaluations. RobustML (`https://www.robust-ml.org/`) was one of the early benchmarks that aimed at collecting robustness claims from the community for the most prominent defenses. Croce et al. (2020) further introduced RobustBench, a popular adversarial robustness benchmark that includes a large repository of models and is based on AutoAttack (Croce and Hein, 2020), an ensemble of four hyperparameter-free attacks (white- and black-box) that has shown reliable performance over a large set of models from the literature. In our work, we follow Croce et al. (2020) and also rely on AutoAttack for standardized measurement of ℓ_∞ adversarial robustness.

4.2 NEURAL SCALING LAWS

Scaling in Deep Learning The exploration into neural scaling laws, which describe how the performance of neural networks changes with varying model size, data, and computational budgets, has been consistently observed and validated across diverse domains like image classification (Bahri et al., 2021; Zhai et al., 2022), language modeling (Kaplan et al., 2020; Sharma and Kaplan, 2022), and neural machine translation (Bansal et al., 2022; Gordon et al., 2021). Recent works have found that scaling up the data size, the model size, and the training schedule often leads to substantially improved performance. Kaplan et al. (2020) elucidates the intricate relationship between model size, dataset size, and computational budget, spotlighting pivotal scaling laws foundational to the efficacy of expansive neural language models. This work also offers strategies to optimally allocate a predetermined compute budget. In a bid to empirically gauge the advantages of scaling, (Alabdulmohsin et al., 2022) introduces a robust methodology to reliably deduce scaling law parameters from learning curves. This methodology, rooted in the *extrapolation loss*, offers insights into how neural architecture variations influence scaling exponents. The work by (Lohn and Musser, 2022) delves deeper into the pivotal role computation plays in AI evolution, presenting insights into potential inflection points where computational power might no longer yield substantial AI advancements.

Environmental Implications The escalating computational requisites of deep learning invariably lead to surges in energy consumption. Strubell et al. (2019) quantifies the approximate financial and environmental costs associated with training language models and proposes recommendations to reduce costs and improve equity in NLP research and practice. Thompson et al. (2020) offers an introspective look at the computational underpinnings of DL models, suggesting that advancements in DL models are increasingly tethered to computational growth which poses economic, technical, and environmental challenges that beckon more efficient computational strategies. Schwartz et al. (2020) emphasizes that the computational expenses of cutting-edge AI endeavors have grown by a factor of 300,000 in recent epochs, resulting in a very substantial carbon footprint. This underscores the imperative to prioritize efficiency as a cardinal metric, aligned with accuracy, to mitigate AI's environmental impact and bolster inclusivity. Echoing these sentiments, Henderson et al. (2020) champions the cause of carbon accountability in AI, introducing a transparent and standardized reporting paradigm. However, similar studies are nearly missing in the context of adversarial robustness. Therefore, in this work, we pioneer an exploration into the neural scaling laws governing adversarial robustness.

5 Metrics of Interest

5.1 Accuracy-Related Metrics

As we are interested in adversarial robustness, we report the AutoAttack accuracy of our models. AutoAttack (Croce and Hein, 2020), an ensemble of four different attacks, is the de-facto standard to benchmark the robustness of deep learning vision models. However, as running AutoAttack is computationally expensive, during training we evaluate the models using PGD-40 on a subset of the test samples, and apply early-stopping based on this metric. This also mitigates potential overfitting of the overall results to AutoAttack.

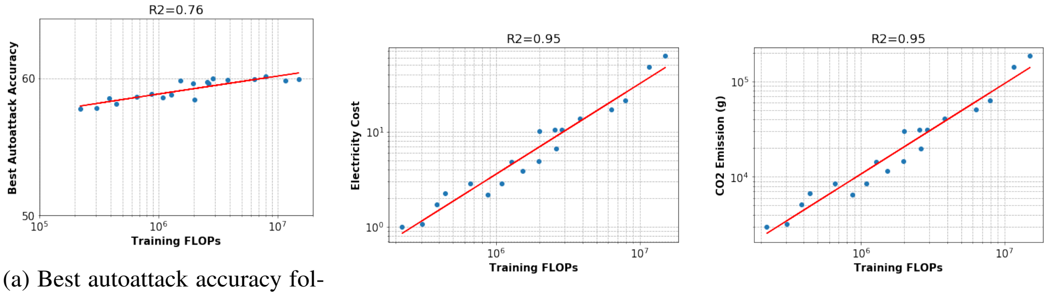
5.2 Efficiency-Related Metrics

To quantify the computational efficiency and environmental cost of adversarial robustness, we perform an analysis of the FLOPs, electricity consumption, and carbon emissions required to train a variety of popular robust models. A detailed description of our methodology is as follows:

FLOPs To gauge model training efficiency in terms of floating-point operations per second (FLOPs), we employ the DeepSpeed Flops Profiler[1]. This measurement is acquired from all submodules of the model throughout the training process. We compute the average total number of FLOPs required for training the model (excluding the validation of the model performed after each epoch), hence accounting for number of epochs, model size, number of attack steps, etc. As DeepSpeed does not enable measuring the FLOPs required for the backward pass, we approximate those FLOPs to be 2× the number of FLOPs needed for the forward pass (Hobbhahn Jsevillamol, 2021).

Electricity Consumption During model training, we repeatedly query the NVIDIA System Management Interface to sample GPU power consumption and get the average over all samples. By averaging these values and integrating them with the training time and the number of GPUs utilized, we estimated the overall power consumption in kilowatt-hours (kWh). Additionally, we incorporated the Power Usage Effectiveness (PUE) coefficient - a metric that encapsulates the ancillary energy expended to sustain the computational infrastructure, primarily cooling. With a reference PUE coefficient of 1.58 (the global average for data centers in 2018), we subsequently computed the cumulative electricity expenditure for training a model, using the U.S. average electricity rate of $0.12/kWh.

[1] DeepSpeed Flops Profiler: https://www.deepspeed.ai/tutorials/flops-profiler/



(a) Best autoattack accuracy follows a power law with a small growth rate.

(b) Electricity cost follows a power law with a high growth rate.

(c) CO2 emission follows a power law with a high growth rate.

Figure 3: Limits of scaling compute on adversarial robustness by analyzing best autoattack accuracy configurations in log-log space.

Carbon Emission To encapsulate the carbon footprint, we adopted the metric of *CO2-equivalents*. This metric facilitates a unified representation of the global warming potential intrinsic to diverse greenhouse gases, expressed as an equivalent quantity of CO_2 (Eggleston et al., 2006). We harnessed the Machine Learning Emissions Calculator (Lacoste et al., 2019) for this purpose, integrating parameters like GPU power consumption (derived from the NVIDIA System Management Interface), model training time, chosen cloud provider, and the geographic locale of computation. For our evaluations, we selected the Google Cloud Platform as the cloud provider and the U.S. as the computational region (with an estimated carbon footprint of 566.3 gCO2/kWh), as shown in Section 3.

6 Are Adversarial Training Techniques Brittle?

Careful readers may have noted, at this point, that the best model among those in Figure 1a has only little more than 60% accuracy (to be precise, 60.14%). They may also have noted that, among the setups we test, there is also the training setup used by Gowal et al. (2021), i.e., WideResNet-70-16 with GELU activation function, 1M synthetic samples, TRADES loss with 10 steps attack, 400 epochs, and EMA. This model has a reported 63.58% AutoAttack accuracy on the original paper, while our model, trained with the same set-up, has 59.92% robust accuracy, more than 3.6% less than the results reported in the paper.

We implemented the adversarial training pipeline from scratch, starting from the `timm` library[2] and by adding components based on the descriptions in the papers we reproduce, as well as by observing the code released with the papers. We also followed the suggestions made by Rade (2021) (e.g., not updating mean and variance of the BatchNorm layers statistics when computing the adversarial example) to match as much as possible the original set-up by (Gowal et al., 2021). We double-checked our implementation and setups, including code, architectures, and hyper-parameters carefully, without any luck in exactly matching the results from Gowal et al. (2021).

While we admit that exactly reproducing those results may not be impossible, it turned out to be very challenging. The fact that it is not possible to reproduce these results suggests that these methods may be relying on relatively brittle features, that are not really robust to small changes in the training setup. Hence, we believe that one factor to keep in account when trying to improve adversarial robustness before scaling up compute is to make sure to rely on setups that are robust to small changes in the training hyperparameters.

7 Discussion and Limitations

Our work presents several limitations, including the fact that we do not match exactly the results from previous work, we are only testing on a low-resolution and small-sized dataset, and we do not have confidence intervals. However, none of these limitations is easy to address in practice: we spent non-negligible time (and compute resources) to reproduce the results from the literature, and scaling

[2] `https://github.com/huggingface/pytorch-image-models`

up to larger datasets such as ImageNet and doing multiple runs per sample would be impractical in terms of compute (training WideResNet-76-10 for 400 epochs with 10-steps TRADES takes more than 16 days on a node with four Nvidia V100 GPUs).

Regardless, we believe that our work still gives a clear overview that scaling up compute does not give as much advantage for adversarial training as it does for standard training, especially without the usage of extra data. Thus, future research should explore the generation and the usage of extra data for improving adversarial training. We advocate for more innovative and comprehensive approaches for continued progress in improving adversarial robustness, encouraging the field to explore new directions. We firmly believe that devising new methods to achieve robust and resilient AI systems is not only necessary but also holds promise for a future where machine learning technologies can withstand adversarial threats.