Revisiting the Adversarial Robustness-Accuracy Tradeoff in Robot Learning

Mathias Lechner, Alexander Amini, *Member, IEEE*, Daniela Rus, *Fellow, IEEE*, and Thomas A. Henzinger

***Abstract*—Adversarial training (i.e., training on adversarially perturbed input data) is a well-studied method for making neural networks robust to potential adversarial attacks during inference. However, the improved robustness does not come for free but rather is accompanied by a decrease in overall model accuracy and performance. Recent work has shown that, in practical robot learning applications, the effects of adversarial training do not pose a fair trade-off but inflict a net loss when measured in holistic robot performance. This work revisits the robustness-accuracy trade-off in robot learning by systematically analyzing if recent advances in robust training methods and theory in conjunction with adversarial robot learning, are capable of making adversarial training suitable for real-world robot applications. We evaluate three different robot learning tasks ranging from autonomous driving in a high-fidelity environment amenable to sim-to-real deployment to mobile robot navigation and gesture recognition. Our results demonstrate that, while these techniques make incremental improvements on the trade-off on a relative scale, the negative impact on the nominal accuracy caused by adversarial training still outweighs the improved robustness by an order of magnitude. We conclude that although progress is happening, further advances in robust learning methods are necessary before they can benefit robot learning tasks in practice.**

***Index Terms*—Deep learning methods, representation learning, transfer learning, robot safety.**

I. Introduction

THIS is the first sentence of my Introduction. Adversarial attacks are well-studied vulnerabilities of deep neural networks [1], [2]. These norm-bounded input perturbations make the network change its decision compared to the unaltered input and can have catastrophic impact in practical robotics applications. Critically, the adversarially altered inputs are barely distinguishable from the original input by humans. Most realistic-sized computer vision networks can be fooled by perturbations that change each pixel by a maximum of 4% (i.e., a l_∞-norm less or equal to 8) while being barely noticeable by humans.

Adversarial robustness is an important consideration in the development of robotic applications, as it ensures that the robot's behavior remains consistent and predictable in the presence of perturbations or attacks. In the real world, robots must be able to operate in a variety of environments and under a wide range of conditions, some of which may be outside of their training data or beyond their control. Adversarial robustness allows robots to continue functioning effectively even when faced with such challenges, improving their reliability and safety in real-world applications. Additionally, as robots become more integrated into society and are given greater autonomy, it becomes increasingly important to ensure that they are not susceptible to manipulation or exploitation by malicious actors. Adversarial robustness helps to protect against such threats and ensure that robots can be trusted to behave in a predictable and responsible manner.

Robust learning aims to tackle the problem by training networks that are immune to adversarial or other types of attacks [3], [4], [5], [6], [7], [8], [9], [10]. One of the most dominant approaches for training robust models is adversarial training which adds adversarial perturbations to the training data online during and throughout the learning procedure [3], [7]. Adversarial training methods improve the test-time robustness on adversarial examples at the critical cost of lower nominal accuracy [11], [12], [13]. For instance, the advanced adversarial training algorithm of [14], which won the NeurIPS 2018 Adversarial Vision Challenge, yielded a robust network with an accuracy of 89% on the CIFAR-10 dataset. In contrast, standard training algorithms can easily produce non-robust networks with an accuracy above 96% on this dataset [15]. This dilemma of choosing between an accurate but vulnerable and a robust but less accurate model is known as the robustness-accuracy trade-off [11], [13], [14].

Recent work [16] has investigated this trade-off specifically in the context of robot learning applications where both accuracy and robustness are critical as the system is ultimately deployed into physical, safety-critical environments. The authors observed that this trade-off is not fair trade but poses a net loss when evaluating the robots' overall performance and concluded that adversarial training is not ready for robot learning. However, recent work has shown that multiple factors (e.g., model size, choice of the activation function, adversarial training procedure)

Manuscript received 18 August 2022; accepted 17 January 2023. Date of publication 31 January 2023; date of current version 8 February 2023. This letter was recommended for publication by Associate Editor N. Figueroa and Editor J. Kober upon evaluation of the reviewers comments. This work was supported in part by the AI2050 Program at Schmidt Futures under Grant G-22-63172, in part by Capgemini SE through Project ERC-2020-AdG under Grant 101020093, in part by the National Science Foundation (NSF), in part by JP Morgan Graduate Fellowships, and in part by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Grant FA8750-19-2-1000. *(Corresponding author: Mathias Lechner.)*

Mathias Lechner, Alexander Amini, and Daniela Rus are with the Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139 USA (e-mail: mlechner@mit.edu; amini@mit.edu; rus@csail.mit.edu).

Thomas A. Henzinger is with the Institute of Science and Technology Austria (ISTA), 3400 Klosterneuburg, Austria (e-mail: thomas.henzinger@ist.ac.at).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3240930, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3240930

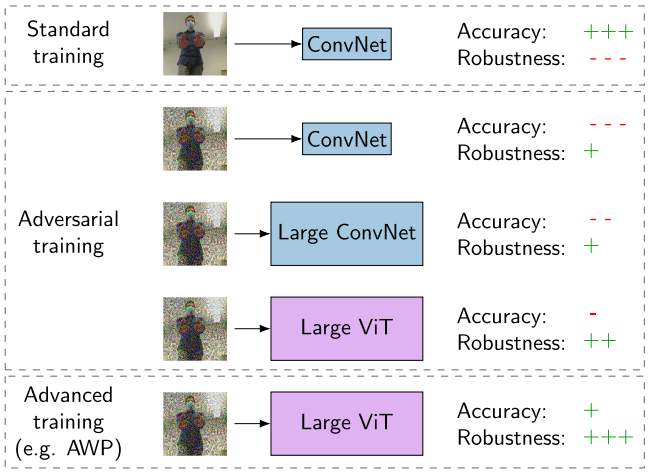

Fig. 1. High-level summary of our results. Adversarial training improves robustness at the cost of significantly reduced accuracy. We show that methods to counteract this decrease in accuracy are most effective when multiple approaches are combined, i.e., an overparametrized network, a vision transformer neural architecture, and advanced adversarial training procedures. Further combined improvements may close the robustness-accuracy gap entirely.

contribute to the reduction in accuracy of robot learning methods [17], [18], [19], [20], [21], [22]. In particular, these works underline that larger models are necessary for robustly fitting the training data [18], [23]. Moreover, they emphasize that a more careful selection of the neural network architecture and hyperparameters is needed when replacing standard training with adversarial training methods [19], [21]. However, there remains a critically important and open question on if these recent advances are sufficient to quell the costs of adversarial training for robotics.

In this work, we assess whether the conclusion of [16] that adversarial training is not ready for robot learning remains true or is challenged by these recent advances in the field. In particular, we evaluate if overparametrized models [18], vision transformers [19], [24], smooth curvature activation functions [20], more careful hyperparameter selection [21], and advanced adversarial training methods [22] can provide acceptable accuracy and robustness on three robot learning and autonomous driving tasks.

Our results show that, although the techniques listed above pose a significant improvement in the robustness-accuracy gap, the negative impact on the nominal accuracy from adversarial training still outweighs the benefits of the induced robustness. Specifically, while the methods from the literature make single digits improvements on the robustness-accuracy Pareto front, i.e., improving both accuracy and robustness, the negative side-effects of adversarial training methods still outweigh these advances by an order of magnitude. Nonetheless, we observed the trend that combining multiple individually introduced robustness enhancement methods provided the most promising future path toward closing the robustness-accuracy gap, e.g., as outlined in Fig. 1.

We summarize our contributions as:

- We evaluate five advancements in robust learning methods in three different real-world robotic applications (456 models tested in total) for their suitability in closing the robustness-accuracy tradeoff gap in robot learning tasks in practice.
- We provide strong empirical evidence that, while robustness can be improved by the methods from literature, the negative effect on the nominal accuracy of adversarial training still outweighs the improvements of these methods by an order of magnitude.
- Our results show that adversarial training is most effective when multiple individual robust learning approaches are combined. This suggests that the most promising path to closing the robustness-accuracy gap entirely in the future is the integration of multiple independent approaches for enhancing robustness.

The remainder of this paper is structured as follows. In Section II, we recapitulate robustness of neural networks, adversarial training, and the robustness-accuracy trade-off. In Section III, we describe related work on improving the robustness of neural networks and avoiding the reduced clean accuracy of adversarial training. Finally, in Section IV, we experimentally evaluate these improvements on three robot learning tasks.

II. Background and Related Work

A neural network is a function $f_\theta : \mathcal{X} \to \mathcal{Y}$ parameterized by θ. In supervised learning, the training objective is to fit the function to a given dataset in the form of $\{(x_1, y_1), \dots (x_n, y_n)\}$ assumed to be i.i.d. sampled from a probability distribution over $\mathcal{X} \times \mathcal{Y}$. This fitting process is done via empirical risk minimization (ERM) that minimizes

$$\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f_\theta(x_i), y_i) \tag{1}$$

via stochastic gradient descent. The differentiable loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ characterizes how well the network's prediction $f_\theta(x_i)$ matches the ground truth label y_i.

An adversarial attack is a sample (x_i, y_i) from the data distribution and a corresponding attack vector μ with $\|\mu\| \leq \varepsilon$ such that $f(x_i) \neq f(x_i + \mu)$ with ε being a threshold. For image data, L_∞ thresholds $\delta \leq 8$ are usually not recognizable or appear as noise for human observers. It has been shown that most neural networks, irrespective of network types, input domains, or learning setting, are susceptible to adversarial attacks [2], [25], [26], [27], [28], [29], [30].

Typical norms used in adversarial attacks are the ℓ_1, ℓ_2, and the ℓ_∞ norm. In this work, we focus on the ℓ_∞ norm. A network is robust on a given sample if no such attack μ exists within a threshold ε. The robust accuracy is the standard metric for measuring the robustness of a network aggregated over an entire dataset $\{(x_1, y_1), \dots (x_n, y_n)\}$ by counting the ratio of correctly classified samples that are also robust.

In practice, deciding whether a network is robust for a sample is an NP-hard problem [31], [32], [33] and, therefore, cannot be computed for typically sized networks in a reasonable time. Instead, the robustness of networks is often studied with respect to empirical gradient and black-box-based attack methods. The fast gradient sign method (FGSM) [2] computes an attack by

$$\mu = \varepsilon \operatorname{sign}\left(\frac{\partial \mathcal{L}(f_\theta(x_i), y_i)}{\partial x_i}\right). \tag{2}$$

Despite its simplicity, adversarial training often uses the FGSM method due to its speed. The iterative fast gradient sign method (I-FGSM) [34] is a more sophisticated generalization of the FSGM. It computes an attack iteratively in k steps starting from $\mu_0 = \mathbf{0}$ and updating it by

$$\mu_i = \frac{\varepsilon}{k} \operatorname{sign}\left(\frac{\partial \mathcal{L}(f_\theta(x_i + \mu_{i-1}), y_i)}{\partial x_i}\right). \tag{3}$$

DeepFool [35], the C&W method [36], and projected gradient descent [7] are other common iterative attack methods that are used for evaluating robustness but are too computationally expensive to incorporate in adversarial training. DeepFool [35] linearizes the network in each iteration of updating μ_i. Projected gradient descent [7] applies unconstrained gradient descent but divides each μ_i by its norm and multiplies the results with ε to project it back into the given threshold. The C&W method [36] avoids such projection by parametrizing the attack vector μ by another variable and a transformation that already normalizes the attack to stay within a given threshold. It has been experimentally shown that any network of non-trivial size is, at least in parts, vulnerable to such attacks [7].

Robust learning methods aim to train networks that are robust [3], [4], [5], [6], [7], [8], [9]. One of the most common robust learning methods is adversarial training which changes the standard ERM objective to the min-max objective

$$\frac{1}{n}\sum_{i=1}^{n} \max_{\mu:\|\mu\|\leq\varepsilon} \mathcal{L}(f_\theta(x_i + \mu), y_i), \tag{4}$$

where $\varepsilon > 0$ is some attack budget controlling how much each input can be perturbed. Due to the computation overhead by this training objective, fast attack-generating methods are typically used for computing the $\max$ in (4), e.g., the FGSM or I-FGSM.

Alternative approaches to adversarial training make minor modifications to the objective term in (4). For instance, the TRADES algorithm [14] replaces the label y_i in (4) with the network's prediction of the original input, i.e., $f_\theta(x_i)$, and optimizes a joint objective of the standard ERM term and the robustness term. The approach of [37] removes the overhead imposed by the maximization step in (4) by pre-computing μ in the previous gradient descent step. Although such pre-computed μ can become inaccurate, i.e., stale, [37] showed that it improves robustness in practice. Adversarial weight perturbation (AWP) [22] improves the generalization of adversarially trained networks by injecting adversarial noise into the weights of the network and smoothing the loss surface. Data augmentation applied to adversarial training has also been shown to positively affect the robustness, and the generalization of neural networks [17]. The work of [38] has shown that the negative impact of adversarial training on the clean accuracy of a network can be further reduced by combining it with advanced data augmentation techniques such as MixUp [39].

The major limitation of adversarial training methods is that they negatively affect the network's standard accuracy (or other performance metrics). For example, medium-sized networks achieve an accuracy of 96% on the CIFAR-10 dataset when trained with standard ERM [15]. However, in [14] the best-performing network trained with the TRADES algorithm could only achieve a standard accuracy of 89% on this dataset. This phenomenon of an antagonistic relation between accuracy and robustness was first studied in [13] and is known as the accuracy-robustness trade-off. The trade-off was studied in the context of robot learning in [16] by investigating whether the gained robustness is worth the reduction in nominal accuracy in real-world robotic tasks. The authors observed that the adversarially trained networks resulted in a worse robot performance than by using a network trained in the standard way.

The concept of adversarial training and the min-max objective of robust learning has been adopted for other task-specific types of specifications, such as safety. For example, [16] has introduced safety-domain training by replacing the norm-bounded neighborhoods of labeled samples with arbitrary sets and corresponding labels, i.e., a min-max training objective over labeled sets. Some modifications of the min-max objective have been studied in feedback systems with closed-loop safety and stability specifications. For instance, [40], [41], [42] propose to learn a safety certificate via a learner-verifier framework where the maximization step is replaced by a verification module that provides formal guarantees on the certificate.

Adversarial training has also been studied as a regularizer for improving the generalization of neural networks. In particular, [43] used mild adversarial attacks based on a hierarchical structure to improve the clean accuracy of vision transformer models [44]. The work of [45] studied human adversaries to improve the performance in robotic object manipulation tasks.

III. Methods

In this section we describe three directions from the literature that point to paths of how to improve robustness without sacrificing standard accuracy.

A. Smooth Activations and Bag of Tricks

Recent work suggests that the common ReLU activation function, i.e., $\max\{0, x\}$, is not well suited for adversarial training methods [20]. Instead, the authors observed that activation functions with smooth curvatures provide better robustness at roughly the same standard accuracy. Specifically, the sigmoid-weighted linear unit (SiLU) activation function [46], i.e., $x \cdot \frac{1}{1+\exp(-x)}$, was highlighted as having a smooth second derivative and observed to improve robustness compared to alternative activations. We note that the SiLU activation was concurrently proposed as swish activation function in [47].

The work of [21] investigated how hyperparameters of the learning process affect adversarial training compared to standard ERM. For example, the authors experiment with learning rate schedules, early stopping, and batch size, among other settings. The authors observed that adversarial training benefits from a higher weight decay factor than standard training. Moreover, the authors confirmed that a smooth activation function improves robustness over the ReLU activation.

B. Robustness Requires Overparametrization

Theoretical contributions to the robustness-accuracy tradeoff recently discovered that overparametrization is necessary for smoothly fitting the training data [23]. While empirical results already suggested that the accuracy of larger models suffers less from adversarial training than for small models, the critical insight is that such large models are necessary. In particular, the authors proved that for a dataset of n samples with d-dimensional features, a model with n parameters can fit the training samples but cannot smoothly interpolate between them. Moreover, the authors show that a model needs at least nd parameters to fit the training data and interpolate them smoothly. The authors also demonstrated that contemporary models for standard datasets do not contain enough parameters with respect to their proven results.

C. Vision Transformers are More Robust Than CNNs

The vision transformer (ViT) [44] is a powerful machine learning architecture that represents an image as a sequence of patches and processes this sequence using a self-attention mechanism [49]. Detailed experimental comparisons between vision transformer and convolutional neural networks suggest that ViTs are naturally more robust with respect to object occlusions and distributions shifts [50]. Concurrent work on comparing ViTs to CNNs with respect to adversarial attacks has found that vision transformers seem to be naturally more robust to adversarial attacks as well.

All advances on the robustness-accuracy tradeoff discussed above are either theoretical or were evaluated on static image classification tasks. Moreover, the methods are typically evaluated on research datasets such as CIFAR and ImageNet. While these datasets allow studying machine learning models' general performance, they significantly differ from real-world robot learning tasks. For example, the CIFAR datasets consist of very low-resolution images, i.e., 32-by-32 pixel, whereas robotic vision processing systems handle images with much higher resolution, e.g., 256-by-256 pixels in [16]. Although the samples of the ImageNet dataset have a realistic image resolution, typical robot learning datasets consist of multiple orders of magnitude fewer samples than the ImageNet dataset. Moreover, experiments on research datasets often report static test metrics, whereas learned robotic controllers are deployed in a closed-loop on a robot.

The next section evaluates the methods described above on multiple real-world robot learning tasks, including open-loop training and closed-loop evaluation on an autonomous driving task.

IV. Experiments

In this section we study the advances in adversarial training methods on three robot learning tasks.

A. End-to-End Driving

Our first experiment considers an autonomous driving task. In particular, a network is trained to predict the curvature of the road ahead of a car from images received at a camera that is mounted on top of the vehicle. The training data is collected by a human driver who maneuvers the car around a test track. The networks are then trained on collected data using supervised learning. Finally, we deploy the networks in a closed-loop autonomous driving simulator. We use the VISTA simulation environment [48] for this purpose.

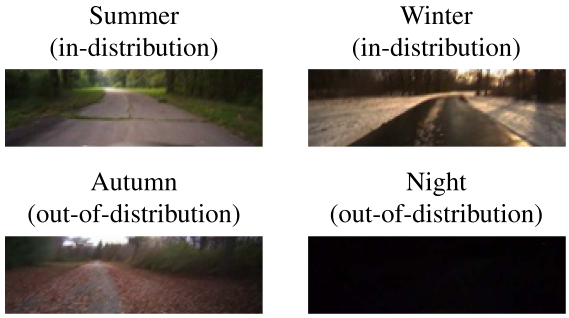

Fig. 2. Test conditions of our closed-loop driving experiment using a data-driven simulation environment [48]. The training data are collected in summer and winter conditions (separated from the testing data).

We compare the performance of a baseline CNN with four variations. First, we compare with an enlarged variant of the baseline CNN to validate the necessity of overparametrization for robustness empirically. Next, we equip the baseline with the smoother SiLU activation and increase the weight decay (wd+). We also test the CNN trained with adversarial weight perturbation (AWP) [22] instead of training via the objective in (4). Finally, we test a vision transformer model. The baseline model (CNN) consists of 440 k, the enlarged model (CNN-large) of 7.7 M, and the tested vision transformer (ViT) of 2.0 M trainable parameters. The inputs of all architectures are 160-by-48 RGB images that are normalized per-image to have zero mean and unit standard deviation. The architecture details of the two convolutional networks are listed in the Supplementary Materials. Our vision transforms splits the input image into non-overlapping patches of 16-by-12 pixels, uses a latent dimension of 256, with 4 attention heads, 384 feed-forward dimensions, and 4 layers in total. For the training, we use the Adam optimizer [51] with a learning rate of 0.0003 and a batch size of 64. The weight decay is set to 10^{-5}, except for the wd+ variant, which is trained with a decay factor of $5 \cdot 10^{-5}$. We train all networks for a total of 900,000 steps. We train all models with standard and adversarial training with increasing attack budget ($\varepsilon = 0, 1, \ldots 8$) and I-FGSM as attack methods.

For each model and attack budget pair, we run a total of 400 simulations, split into 200 in-training distribution, and 200 out-of-training distribution condition runs. The in-training data were collected in summer and winter and were separated from the training data, i.e., there is no overlap between the training data and the evaluation data. The out-of-training data were collected in autumn and during the night, with no such condition present in the training data. The four conditions are visualized in Fig. 2. As an evaluation metric, we report the number of crashes during the simulation, i.e., when the vehicle leaves the road.

TABLE I
ROBUST VALIDATION ACCURACY (UNDER I-FGSM WITH $\varepsilon = 8$) AND TEST ACCURACY ON THE VISUAL GESTURE RECOGNITION DATASET OF VARIOUS ADVERSARIALLY FINE-TUNED MODELS. BEST ROBUST VALIDATION ACCURACY AND TEST ACCURACIES GREATER THAN 80% ARE HIGHLIGHTED IN BOLD. BEST VALUES ARE UNDERLINED

Model	Adversarial training budget	Robust validation accuracy	Test accuracy
ResNet50	$\varepsilon = 0$	2.3% ± 1.9	**93.5%** ± 3.8
	$\varepsilon = 1$	18.9% ± 3.2	**86.1%** ± 3.1
	$\varepsilon = 2$	55.3% ± 3.1	76.0% ± 2.5
	$\varepsilon = 4$	77.0% ± 3.8	68.6% ± 3.2
	$\varepsilon = 8$	60.7% ± 2.7	51.8% ± 6.6
	$\varepsilon = 4$ (+AWP)	75.0% ± 1.0	67.7% ± 8.6
	$\varepsilon = 8$ (+AWP)	47.6% ± 5.5	41.4% ± 0.4
ResNet101	$\varepsilon = 0$	3.8% ± 1.3	**85.9%** ± 7.3
	$\varepsilon = 1$	20.7% ± 2.2	**82.9%** ± 4.4
	$\varepsilon = 2$	54.8% ± 1.4	76.1% ± 2.6
	$\varepsilon = 4$	44.4% ± 0.2	41.5% ± 1.2
	$\varepsilon = 8$	43.9% ± 0.5	41.7% ± 0.0
	$\varepsilon = 4$ (+AWP)	43.8% ± 0.4	41.7% ± 0.0
	$\varepsilon = 8$ (+AWP)	44.2% ± 0.4	41.7% ± 0.0
ResNet152	$\varepsilon = 0$	4.4% ± 4.4	**87.2%** ± 4.2
	$\varepsilon = 1$	29.6% ± 5.7	**82.1%** ± 7.0
	$\varepsilon = 2$	66.7% ± 4.2	75.8% ± 4.4
	$\varepsilon = 4$	52.6% ± 9.8	50.8% ± 12.4
	$\varepsilon = 8$	44.5% ± 0.4	41.7% ± 0.0
	$\varepsilon = 4$ (+AWP)	**78.0%** ± 3.5	67.3% ± 1.6
	$\varepsilon = 8$ (+AWP)	44.2% ± 0.4	41.7% ± 0.0

TABLE II
ROBUST VALIDATION ACCURACY (UNDER I-FGSM WITH $\varepsilon = 8$) AND TEST ACCURACY ON THE VISUAL GESTURE RECOGNITION DATASET OF VARIOUS ADVERSARIALLY FINE-TUNED MODELS. BEST ROBUST VALIDATION ACCURACY AND TEST ACCURACIES GREATER THAN 80% ARE HIGHLIGHTED IN BOLD. BEST VALUES ARE UNDERLINED

Model	Adversarial training budget	Robust validation accuracy	Test accuracy
ViT-Small/16	$\varepsilon = 0$	0.8% ± 1.5	**94.1%** ± 2.1
	$\varepsilon = 1$	21.3% ± 5.6	**84.6%** ± 6.0
	$\varepsilon = 2$	48.0% ± 4.8	**83.3%** ± 4.7
	$\varepsilon = 4$	72.3% ± 5.0	75.2% ± 3.1
	$\varepsilon = 8$	43.9% ± 0.4	41.1% ± 1.9
	$\varepsilon = 4$ (+AWP)	58.3% ± 2.0	**85.5%** ± 0.4
	$\varepsilon = 8$ (+AWP)	61.4% ± 1.8	66.6% ± 3.7
ViT-Base/16	$\varepsilon = 0$	11.7% ± 4.5	**82.3%** ± 5.7
	$\varepsilon = 1$	37.9% ± 8.4	79.4% ± 1.7
	$\varepsilon = 2$	61.4% ± 3.8	**84.5%** ± 4.6
	$\varepsilon = 4$	67.1% ± 11.7	76.0% ± 1.9
	$\varepsilon = 8$	48.0% ± 3.7	54.6% ± 10.2
	$\varepsilon = 4$ (+AWP)	59.0% ± 9.5	84.6% ± 1.9
	$\varepsilon = 8$ (+AWP)	76.0% ± 11.0	80.0% ± 9.5
ViT-Large/16	$\varepsilon = 0$	20.8% ± 11.0	**85.9%** ± 2.5
	$\varepsilon = 1$	35.3% ± 1.2	**89.9%** ± 3.6
	$\varepsilon = 2$	67.1% ± 9.9	**89.6%** ± 3.2
	$\varepsilon = 4$	77.4% ± 6.6	71.0% ± 15.9
	$\varepsilon = 8$	58.0% ± 17.5	47.2% ± 9.8
	$\varepsilon = 4$ (+AWP)	73.9% ± 7.3	**86.3%** ± 7.4
	$\varepsilon = 8$ (+AWP)	**89.7%** ± 0.6	**88.9%** ± 3.1

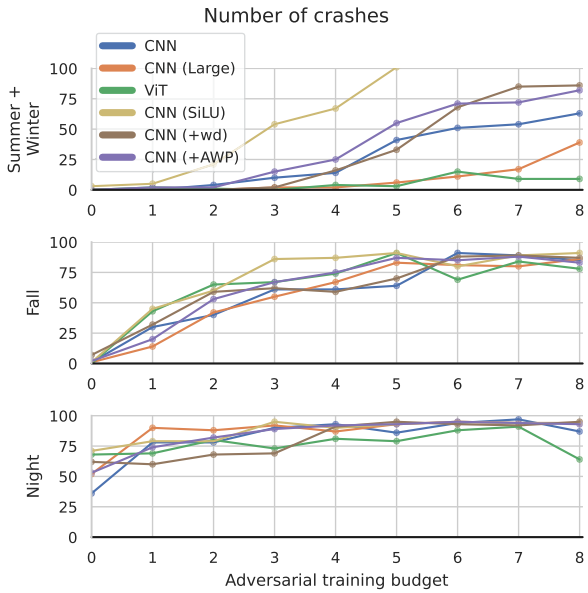

Fig. 3. Number of crashes out of 100 simulation runs in each data setting (summer, winter, fall, night) with respect to varying the adversarial training budget. All models were trained in summer and winter conditions (on a different data split than the evaluations). The large CNN and the ViT model perform best under heavy adversarial training, but no adversarially trained model could handle distribution shifts, i.e., fall and night conditions.

The top row in Fig. 3 shows the crashes during the summer and winter simulations. The results show that the overparametrized model and the vision transformer indeed provide better performance at a larger adversarial training budget than the baseline. An increased weight decay improved the performance only at lower attack budget training, while the networks with SiLU activation performed worse in the closed-loop tests. At larger attack budgets, no model could drive the car safely, while most models learned by standard ERM could drive all 200 runs flawlessly.

The out-of-training distribution simulation results for autumn and night conditions are shown in the middle and bottom row in Fig. 3. We observe that adversarial training significantly hurt the out-of-distribution performance of all models, i.e., especially in the autumn data. A video demonstration of the simulated runs is available at https://youtu.be/TQKP7l9PfNo. In summary, the best driving performance across all four tested conditions was observed with networks trained with standard ERM.

B. Visual Gesture Recognition

Our second experiment concerns training an image classifier that controls the operating modes of a mobile robot as reported in [16]. The dataset consists of 2029 sample 256-by-256 pixel images corresponding to three classes, i.e., idle (905 samples), enable (552 samples), and disable (572 samples), which are split into a training and a validation set with a 90%:10% ratio. The experiments on the physical robot in [16] suggest that a validation accuracy of above 90% is necessary for acceptable robot performance. Due to the small size of the dataset, we resort to transfer learning of a pre-trained classifier using the big-transfer (BiT) fine-tuning protocol of initializing the output layer with all zeros and training all layers [52].

In this experiment, we test the theoretical necessity of overparametrization in practice. We train networks of different sizes

TABLE III
VALIDATION ACCURACY ON THE ROBOT FOLLOW DATASET [16] OF 1D-CONVOLUTIONAL NNS WITH VARIOUS HYPERPARAMETERS AND TRAINED WITH STANDARD AND SAFETY-DOMAIN TRAINING. VALUES GREATER THAN 80% ARE HIGHLIGHTED IN BOLD. SAFETY LEVEL 0 CORRESPONDS TO STANDARD TRAINING, WHILE THE NETWORK TRAINED WITH SAFETY LEVEL 1 AND ABOVE PROVIDE FORMAL SAFETY GUARANTEES OF NEVER CRASHING THE ROBOT INTO AN OBSTACLE. THE COLUMNS SHOW NETWORKS WITH DIFFERENT WIDENING FACTOR. THE NUMBER OF LEARNABLE PARAMETERS ARE SHOWN IN PARENTHESIS

Safety level		Validation accuracy Width 1 (360k)	Width 2 (1.4M)	Width 3 (3.2M)	Width 4 (5.7M)
0	Baseline	**83.2%** ± 0.8	**84.7%** ± 1.6	**83.9%** ± 1.9	**85.2%** ± 0.8
	ELU	73.3% ± 1.5	72.5% ± 3.3	73.3% ± 0.8	71.3% ± 1.3
	wd+	**82.5%** ± 2.0	**84.0%** ± 2.1	**85.7%** ± 1.3	**85.7%** ± 1.2
1	Baseline	75.1% ± 2.6	78.6% ± 3.7	77.4% ± 2.1	78.7% ± 3.4
	ELU	53.1% ± 0.6	53.5% ± 0.4	52.9% ± 0.6	52.3% ± 0.8
	wd+	74.2% ± 3.4	75.0% ± 1.8	65.9% ± 10.7	67.4% ± 12.0
2	Baseline	76.3% ± 3.1	76.8% ± 4.9	76.1% ± 2.8	78.5% ± 3.2
	ELU	53.6% ± 0.3	53.1% ± 0.3	53.2% ± 0.4	52.9% ± 0.6
	wd+	72.9% ± 3.3	75.5% ± 2.1	68.4% ± 8.6	70.7% ± 10.0
3	Baseline	51.8% ± 0.9	52.8% ± 0.5	53.3% ± 0.1	53.9% ± 0.3
	ELU	53.2% ± 0.8	53.8% ± 0.5	53.1% ± 0.1	53.2% ± 0.4
	wd+	51.4% ± 1.1	52.8% ± 0.7	52.8% ± 0.6	53.4% ± 0.4

using adversarial training with increasing attack budget ($\varepsilon \in \{0, 1, 2, 4, 8\}$) and report the robust validation accuracy under I-FGSM attacks with a radius of $\varepsilon = 8$. We also evaluate models trained with adversarial weight perturbation (AWP) [22] and $\varepsilon \in \{4, 8\}$.

As a proxy for real-world test accuracy, we collect a new dataset comprising 190 idle samples, 129 enable samples, and 140 disable samples. Particularly, the test set resembles a real-world deployment of the model on the robot and ensures that there is no spurious temporal or spatial correlation with the original data source. We use the clean accuracy of the new set as our test metric to estimate real-world performance.

For increasing the size of the model, we test a ResNet50 (24 M), ResNet101 (43 M), and ResNet152 (58 M) with the number of trainable parameters reported in parenthesis [53]. We also evaluate the vision transformer models ViT-Small (22 M), ViT-Base (86 M), and ViT-Large (304 M) that process the images in the form of 16-by-16 pixel patches [44]. For the training, we use the Adam optimizer [51] with a learning rate of 0.00005 and a batch size of 64, except for the ResNet152 where a batch size of 32 is used due to out-of-memory errors. We repeat each training run with 5 random seeds and report the mean and standard deviation.

The results in Tables I and II show that the overall best test accuracy could be achieved with standard empirical risk minimization and a ResNet50 or ViT-Small model. As expected, however, these models provide no robustness to adversarial attacks. Still, acceptable test performance ($\geq$80%) at non-trivial robustness was realized by models trained with a small attack budget, e.g. $\varepsilon = \{1, 2\}$. Nonetheless, the gap between the overall best test accuracy and the top-scoring adversarially trained models is significant, i.e., over one and two standard deviations of the standard trained ResNet50 and ViT-Small model, respectively.

The results in Tables I and II show the trend that with an increase in model size, the models become more accurate under adversarial training. This effect is even more amplified when considering the more advanced adversarial weight perturbation training (+AWP). Specifically, the most robust ResNet and vision transformer are both their largest variant trained with AWP. Moreover, we observe an advantage of the ViT architecture over the ResNets in terms of robustness, which has been studied in more detail in [19], [24]. This result suggests that even larger ViT-based models combined with even more advanced adversarial training schemes may be able to close the robustness-accuracy gap entirely.

C. Certified Safety-Domain Training

Adversarial training methods do not ensure robustness but provide only empirical improvements over common attack methods. Certified training methods such as the interval bound propagation [54] can learn networks with formal robustness or safety guarantees. In this experiment, we study the safety-domain training of LiDAR-based mobile robot navigation controller [16]. The objective of the learned controller is to map 541-dimensional laser range scans to 7 possible categories, i.e., stay, straight forward, left forward, right forward, straight backward, left backward, and right backward. The dataset consists of 2705 training and 570 validation samples uniformly distributed across the seven classes. Using safety-domain training, we want to ensure that the robot never crashes into an object in front of it. This is achieved by training an abstract interpretation representation of the network to never output a forward locomotion class in case the LiDAR input indicates an obstacle. There are four safety levels with different strictness of what accounts for an obstacle, e.g., several consecutive rays or just a single ray, defined in [16]. Safety level 0 corresponds to standard training, while safety level 3 represents the strictest level.

We test the overparametrization, increased weight decay (from 0 to 10^{-5}), and smooth activation function methods on this task. As a baseline, we use the 1D-CNN from [16], which is comprised of 360 k parameters. Our overparametrized models increase the width of the network to obtain CNNs with 1.4 M, 3.2 M, and 5.7 M parameters respectively. We use the exponential linear unit (ELU) activation function [55] to represent a smooth activation due to the non-monotonicity of SiLU being

incompatible with the used abstract interpretation domains. We train all models with the Adam optimizer [51] with a learning rate of 0.0001 and a batch size of 64. The safety level 0 models are trained for 20 epochs, while the networks trained using safety-domain training for 2000 epochs. The network architectures are shown in the Supplementary Materials.

We report the validation accuracy as an evaluation metric. The experiments on the physical robot in [16] suggest that a validation accuracy above 80% is necessary to achieve an acceptable real-world performance. Note that all models, except those trained with safety level 0, provide some form of formal safety guarantees. Therefore, this experiment studies how much validation accuracy is traded for the ensured safety. We repeat each training run with 5 random seeds and report the mean and standard deviation.

The result in Table III shows that safety-domain training benefits from an increased number of parameters (width). However, the improvement over the baseline is rather incremental and accounts only for a few percent. In contrast, the accuracy reduction caused by the safety-domain training is several times more significant, e.g., around 10%, and no network trained with safety-domain training exceeds the threshold of 80% accuracy. The networks with smooth activation function and increased weight decay performed worse than the baseline when using safety-domain training. This suggests that certified training methods such as safety-domain training may require different hyperparameters and learning settings than adversarial training.

V. Discussion and Conclusion

Adversarial training (i.e., training on adversarially perturbed input data) is a well-studied method for making neural networks robust to potential adversarial attacks during inference. However, the improved robustness does not come for free but rather is accompanied by a decrease in nominal model accuracy and performance [14]. Recent work [16] has shown that, in practical robot learning applications, the effects of adversarial training do not pose a fair trade-off but inflict a net loss when measured in holistic robot performance. This work revisited the robustness-accuracy trade-off in robot learning by systematically analyzing if recent advances in robust training methods and theory in conjunction with adversarial robot learning can make adversarial training suitable for real-world robot applications.

We evaluated a total of five robust training methods on three different robot learning tasks ranging from autonomous driving in a high-fidelity environment amenable to sim-to-real deployment to mobile robot navigation and gesture recognition. Our results indicate that the negative impact on the nominal accuracy from adversarial training still outweighs the induced robustness. In other words, while adversarial training can improve the model's ability to withstand attacks, it does not justify the reduced accuracy on clean, non-adversarial data.

Nonetheless, our results suggest that, in aggregate, when combining these methods, a significant improvement in the robustness-accuracy gap is made. For instance, the combination of overparametrization, a vision transformer, and a more advanced training scheme (adversarial weight perturbation) performs much better under adversarial training than the models tested in [16]. This suggests that future research directions that can be further combined, e.g., data augmentation or other training schemes, may be able to close the robustness-accuracy entirely.

Acknowledgment

We thank Christoph Lampert for inspiring this work. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

[1] C. Szegedy et al., "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Representations*, 2014.
[2] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2015.
[3] A. Wald, "Statistical decision functions which minimize the maximum risk," *Ann. Math.*, vol. 46, pp. 265–280, 1945.
[4] P. J. Huber, "Robust estimation of a location parameter," *Ann. Math. Statist.*, vol. 1, no. 35, pp. 492–518, 1964.
[5] C. G. Atkeson and S. Schaal, "Robot learning from demonstration," in *Proc. Int. Conf. Mach. Learn.*, 1997, pp. 12–20.
[6] H. Xu, C. Caramanis, and S. Mannor, "Robustness and regularization of support vector machines," *J. Mach. Learn. Res.*, vol. 10, pp. 1485–1510, 2009.
[7] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Representations*, 2018.
[8] C. Song, K. He, L. Wang, and J. E. Hopcroft, "Improving the generalization of adversarial training with domain adaptation," in *Proc. Int. Conf. Learn. Representations*, 2019.
[9] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 4582–4587.
[10] N. Konstantinov and C. Lampert, "Robust learning from untrusted sources," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3488–3498.
[11] A. Raghunathan, S. M. Xie, F. Yang, J. Duchi, and P. Liang, "Adversarial training can hurt generalization," *CoRR*, vol. abs/1906.06032.
[12] H. Zhang, T.-W. Weng, P.-Y. Chen, C.-J. Hsieh, and L. Daniel, "Efficient neural network robustness certification with general activation functions," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 944–4953.
[13] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *Proc. Int. Conf. Learn. Representations*, 2018.
[14] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7472–7482.
[15] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 4140–4147.
[16] M. Lechner, R. M. Hasani, R. Grosu, D. Rus, and T. A. Henzinger, "Adversarial training is not ready for robot learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 4140–4147.
[17] S.-A. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. A. Mann, "Data augmentation can improve robustness," in *Proc. Conf. Neural Inf. Process. Syst.*, 2021, pp. 29935–29948.
[18] S. Bubeck, Y. Li, and D. M. Nagaraj, "A law of robustness for two-layers neural networks," in *Proc. Conf. Learn. Theory*, 2021, pp. 804–820.
[19] D. Zhou et al., "Understanding the robustness in vision transformers," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 27378–27394.
[20] V. Singla, S. Singla, S. Feizi, and D. Jacobs, "Low curvature activations reduce overfitting in adversarial training," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 16403–16413.

[21] T. Pang, X. Yang, Y. Dong, H. Su, and J. Zhu, "Bag of tricks for adversarial training," in *Proc. Int. Conf. Learn. Representations*, 2021.
[22] D. Wu, S.-T. Xia, and Y. Wang, "Adversarial weight perturbation helps robust generalization," in *Proc. Conf. Neural Inf. Process. Syst.*, 2020, pp. 2958–2969.
[23] S. Bubeck and M. Sellke, "A universal law of robustness via isoperimetry," in *Proc. Conf. Neural Inf. Process. Syst.*, 2021, pp. 28811–28822.
[24] S. Paul and P.-Y. Chen, "Vision transformers are robust learners," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 2071–2081.
[25] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," in *Proc. NeurIPS Mach. Learn. Comput. Secur. Workshop*, 2017.
[26] J. Uesato, B. O'Donoghue, A.V. D. Oord, and P. Kohli, "Adversarial risk and the dangers of evaluating against weak attacks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5025–5034.
[27] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2018.
[28] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *Proc. Annu. Netw. Distrib. Syst. Secur. Symp.*, 2019.
[29] M. Giacobbe, T. A. Henzinger, and M. Lechner, "How many bits does it take to quantize your neural network?," in *Proc. Int. Conf. Tools Algorithms Construction Anal. Syst.*, 2020, pp. 79–97.
[30] Y. Bai, J. Mei, A. L. Yuille, and C. Xie, "Are transformers more robust than CNNs?," in *Proc. Conf. Neural Inf. Process. Syst.*, 2021, pp. 26831–26843.
[31] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Proc. Int. Conf. Comput. Aided Verification*, 2017, pp. 97–117.
[32] M. Sälzer and M. Lange, "Reachability is NP-complete even for the simplest neural networks," in *Proc. Int. Conf. Reachability Prob.*, 2021, pp. 149–164.
[33] T. A. Henzinger, M. Lechner, and Žikelić, "Scalable verification of quantized neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 3787–3795.
[34] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. Int. Conf. Learn. Representations*, 2017.
[35] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A simple and accurate method to fool deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2574–2582.
[36] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.
[37] A. Shafahi et al., "Adversarial training for free!," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 3353–3364.
[38] A. Lamb et al., "Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy," *Neural Netw.*, vol. 154, pp. 218–233, 2022.
[39] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *Proc. Int. Conf. Learn. Representations*, 2018.
[40] Y. Chang, N. Roohi, and S. Gao, "Neural Lyapunov control," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 3240–3249.
[41] M. Lechner, D. Zikelic, K. Chatterjee, and T. A. Henzinger, "Infinite time horizon safety of Bayesian neural networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2021, pp. 5998–6008.
[42] M. Lechner, K. Žikelić Chatterjee, and T. A. Henzinger, "Stability verification in stochastic control systems via neural network supermartingales," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 7326–7336.
[43] C. Herrmann et al., "Pyramid adversarial training improves VIT performance," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13419–13429.
[44] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021.
[45] J. Duan, Q. Wang, L. Pinto, C.-C. J. Kuo, and S. Nikolaidis, "Robot learning via human adversarial games," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1056–1063.
[46] S. Elfwing, E. Uchibe, and K. Doya, "Sigmoid-weighted linear units for neural network function approximation in reinforcement learning," *Neural Netw.*, vol. 107, pp. 3–11, 2018.
[47] P. Ramachandran, B. Zoph, and Q. V. Le, "Searching for activation functions," in *Proc. Int. Conf. Learn. Representations, Workshop Track*, 2018.
[48] A. Amini et al., "Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, *arXiv:2111.12083*.
[49] A. Vaswani et al., "Attention is all you need," in *Proc. Conf. Neural Inf. Process. Syst.*, 2017.
[50] M. Naseer, K. Ranasinghe, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Intriguing properties of vision transformers," in *Proc. Conf. Neural Inf. Process. Syst.*, 2021, pp. 23296–23308.
[51] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.
[52] A. Kolesnikov et al., "Big transfer (BiT): General visual representation learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 491–507.
[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[54] S. Gowal et al., "Scalable verified training for provably robust image classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4841–4850.
[55] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. Int. Conf. Learn. Representations*, 2016.