Adversarially Robust Continual Learning

Hikmat Khan
Rowan University
Glassboro, New Jersey, USA
khanhi83@students.rowan.edu

Nidhal Carla Bouaynaya
Rowan University
Glassboro, New Jersey, USA
bouaynaya@rowan.edu

Ghulam Rasool
Moffitt Cancer Center
Tampa, Florida, USA
ghulam.rasool@moffitt.org

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892970

***Abstract*—Recent approaches in continual learning (CL) have focused on extracting various types of features from multi-task datasets to prevent catastrophic forgetting — without formally evaluating the quality, robustness and usefulness of these features. Recently, it has been shown that adversarial robustness can be understood by decomposing learned features into robust and non-robust types. The robust features have been used to build robust datasets and have been shown to increase adversarial robustness significantly. There has not been any assessment on using such robust features in CL frameworks to enhance the robustness of CL models against adversarial attacks. Current CL algorithms use standard features - a mixture of robust and non-robust features - and result in models vulnerable to both natural and adversarial noise.**

This paper presents an empirical study to demonstrate the importance of robust features in the context of class incremental learning (CIL). We adopted the publicly available CIFAR10 dataset for our CIL experiments. We used CIFAR10-Corrupted dataset to evaluate the robustness of the standard, robust and non-robust models against various types of noise including brightness, contrast, Gaussian noise and more. To test these models against adversarially attacked input, we created a new dataset using the project gradient descent (PGD) and fast gradient sign (FGSM) algorithm.

Our experiments demonstrate that a set of models trained on the standard (a mixture of both robust and non-robust) features obtained a higher accuracy compared to the models trained either using robust features or non-robust features. However, the models trained using standard and non-robust features performed poorly in noisy and adversarial conditions as compared to the model trained using robust features. The model trained using non-robust features performed the worst in noisy conditions and under adversarial attacks. Our study underlines the significance of using robust features in CIL.

I. Introduction

Continual learning is an active and challenging area of research [1], [2]. One of the most challenging aspects of continual learning is the catastrophic forgetting phenomenon, in which a model experiences rapid performance degradation on past tasks while it focuses on learning the current task [3], [4]. The model has to strike a balance between learning the classification-related features of the current task and the previous task, which is also known as the *plasticity-stability* dilemma [5]. Current approaches to mitigate catastrophic forgetting can broadly be categorized into five groups [6]. The first is *regularization-based methods*; these methods define metrics to measure the important features and constraint them from being changed in efforts to prevent the model from forgetting previous tasks [7], [8], [9], [10], [11], [12]. The second is *rehearsal-based methods*; these methods play features of the previous task from the memory buffer while learning the features of the current task, simultaneously [13], [14], [15], [16], [17]. The third is a *pseudo-rehearsal based methods*; in this group, a generator is used to create synthetic features from the previous task to be replayed for the model, while learning the original features of the current task [18], [19], [20]. The fourth is *architectural-based methods*; these methods freeze a portion of the network which has learned the current task-specific features and keeps the model from forgetting them [21], [22], [23], [24], [25]. It keeps freezing parts of the network as it learns more tasks, and continues until the model reaches its capacity. The fifth is *saliency-based methods*; these methods use saliency maps of the previous tasks in the replay buffer while the model is learning the current task [26], [27]. The saliency maps are created using activation map algorithms, which capture the important features of the previous task [27], [28], [29], [30], [31], [32].

All the above methods either focus on learning the current task features or on replaying previous features simultaneously, to prevent the model from catastrophic forgetting; however, the question is whether the quality of the input features itself plays a significant role in the continual learning [33], [34]. We took inspiration from the field of adversarial machine learning where it has been established that the extracted features can be classified into robust features and non-robust features [35], [36]. Non-robust features carry transferable/shared knowledge between classes and robust features carry class-specific information [35]. It has been proven that non-robust models are susceptible to adversarial attacks where robust features are stronger and not as easily attacked [35]. Adversarial machine learning has clearly defined categories of features (robust and non-robust) but in continual learning, no such framework has been defined [35]. They further have shown how a dataset can be separated into two datasets, one containing robust features and the other containing non-robust features [35]. In this paper, we explore the role of these two groups of features in continual learning. We consider three versions of a dataset (standard, robust, non-robust dataset) and train five continual learning models on them in class incremental settings. We train them on different types of features to analyze how they react to various types of noise including adversarial. Based upon our analysis, we found that models trained on robust features are more suitable to be deployed in real-world scenarios as compared to models trained on non-robust features.

978-1-7281-8671-9/22/$31.00 ©2022 IEEE

In summary, our contributions are three-fold:

- We have shown that regardless of which data-set (standard, robust, non-robust) the model is trained on, it was able to obtain comparable accuracy to that of the standard model.
- The features of the input are directly related to how strong a model is against noise and adversarial attacks. Better quality input features lead to a better performing model; it can resist natural as well as worst-case adversarial perturbations.
- Models which are trained on non-robust features experience the most degradation. Such degraded performance limits their applicability to be deployed in safety-critical systems.

II. Adversarially Robust and non-Robust Features

Consider a classification problem with input-label pairs $(x, y) \in \mathcal{X}$ sampled from a distribution $\mathcal{D}$. A feature f is defined as a function mapping from the input space $\mathcal{X}$ to the real numbers, with the set of all features thus being $\mathcal{F} = \{f : \mathcal{X} \mapsto \mathbb{R}\}$. Andrew *et al.* proposed a framework for disentangling adversarially robust and non-robust features [35]. A feature f is ρ-useful ($\rho > 0$) if it is correlated with the true label in expectation, that is if

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[y.f(x)] \geq \rho. \tag{1}$$

Suppose we have a ρ-useful feature f. We refer to f as a *robust feature* - formally a γ-robustly useful feature for $\gamma > 0$ - if, under adversarial perturbation (for some specified set of valid perturbations Δ), f remains γ-useful. Formally, we have

$$\mathbb{E}_{(x,y)\sim\mathcal{D}}[\inf_{\delta\in\Delta(x)} y.f(x+\delta)] \geq \gamma. \tag{2}$$

The robust features are extracted by constructing a distribution $\widehat{\mathcal{D}}_R$ which satisfies:

$$\mathbb{E}_{(x,y)\sim\widehat{\mathcal{D}}_R}[y.f(x)] = \begin{cases} \mathbb{E}_{(x,y)\sim\mathcal{D}}[y.f(x)] & \text{if } f \in \mathcal{F} \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $\mathcal{F}$ represents the set of features utilized by the model classifier. Conceptually, we want the robust features to be as useful as they were in the original distribution $\mathcal{D}$ while ensuring that the remaining features are not useful under $\widehat{\mathcal{D}}_{NR}$. The training set for $\widehat{\mathcal{D}}_R$ is constructed via a one-to-one mapping $x \mapsto x_r$ from the original training set for $\mathcal{D}$. We used the optimization framework in [35] to disentangle the standard CIFAR10($\mathcal{D}$) dataset into robust CIFAR10($\mathcal{R}$) and non-robust CIFAR10($\mathcal{N}$) versions. Robust features are those that are more descriptive, distinctive and carry more class-specific information. Non-robust features are spurious, generic and imperceptible. Figures 1, 2 and 3 present the visual examples of standard, robust and non-robust datasets, respectively.

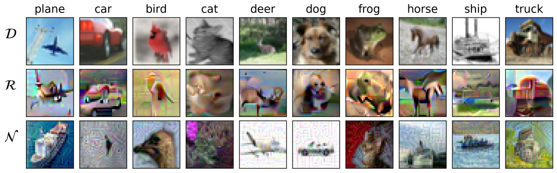

Fig. 1. Standard, Robust and Non-robust CIPHAR-10 datasets. Each column represents a class. 1st row: Sample images from standard CIFAR10($\mathcal{D}$) for all 10 classes. 2nd row: The robustified sample images from robust CIFAR10($\mathcal{R}$). 3rd row: Samples images from non-robust CIFAR10($\mathcal{N}$) dataset.

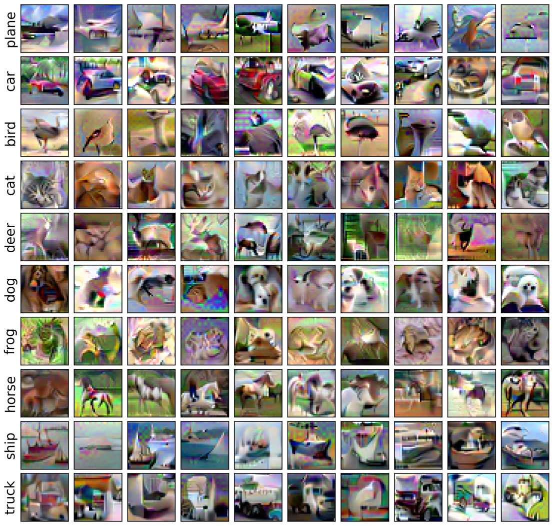

Fig. 2. Sample images from the robustified CIFAR10($\mathcal{R}$) dataset for ten classes. Each row represents a class. Each column showcases sample robust images from each class. For example, row 1 represents the "airplane" class, and each column has a different robust image belonging to this class. Note that the robust features are sparse and carry class-specific information.

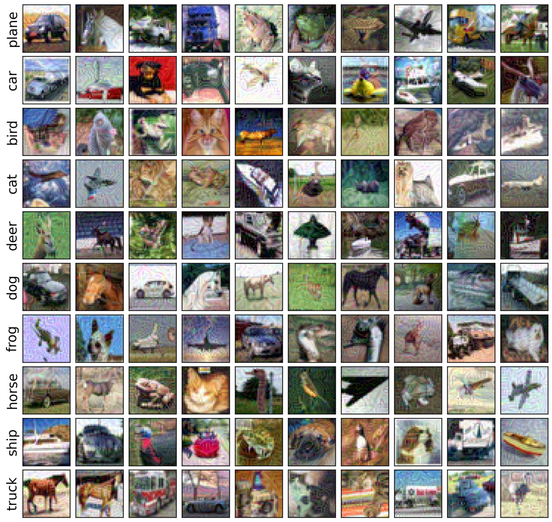

Fig. 3. Sample images from the non-robustified CIFAR10($\mathcal{N}$) dataset for ten classes. Each row represents a class. Each column showcases sample non-robust images from each class. For example, row 1 represents the "airplane" class, and each column has a different non-robust image belonging to this class. Observe that the non-robust features are generic and class-transferable.

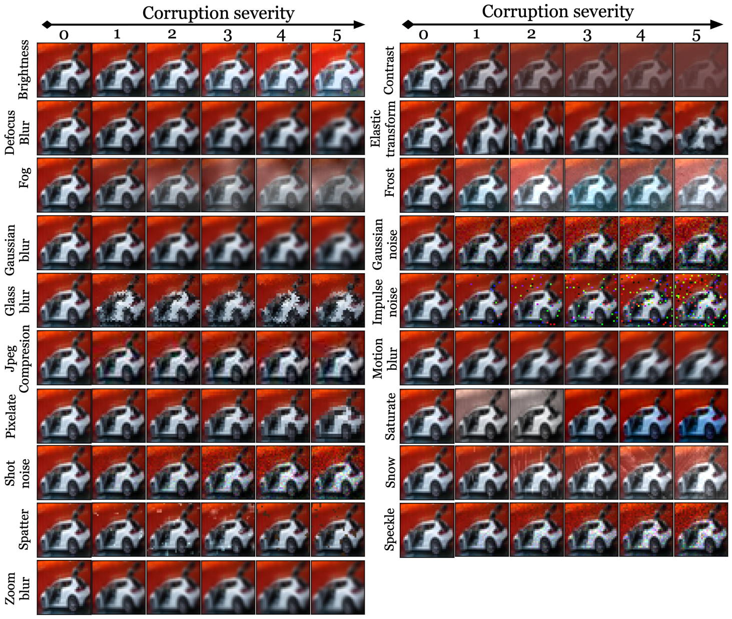

Fig. 4. Examples of the six different severity levels for the 19 different noise types. Severity 0 represents the original clean image whereas severity 5 corresponds to the most severe corruption (Best viewed in color).

III. Benchmarking Neural Network Robustness

A. Common Corruptions (CIFAR10-Corruption) Dataset

Hendrycks *et al.* established rigorous robustness benchmarks, which standardize and broaden the issue of corruption robustness for image classification [37]. Their benchmarks can be used to assess performance of neural networks on common corruptions and perturbations, which can be helpful in identifying networks that can be robustly generalized. The CIFAR10-Corrupted dataset from their benchmarks, contains 19 different forms of algorithmically generated corruptions from noise, blur, weather, and digital categories. Each corruption includes five levels of severity with varying intensities. For instance, there are five levels of severity for blur. Examples of these five levels of severity on the 19 types of noise are shown in Figure 4. The example with severity level zero has the least corruption, while the one with severity level five has the highest level of corruption.

We utilize CIFAR10-C to examine the robustness and sensitivity of the CL models trained in the class incremental learning (CIL) setting against these 19 forms of noise. We utilized the CIFAR10-C dataset to assess (i.e, to evaluate empirically) the impact of the features on which the model was trained. We found that the model that does not base its learning on robust features throughout the learning phase would suffer the most (see the result Section). Furthermore, testing the CL models on such a dataset would emphasize the algorithm's capability/applicability for use in safety-critical applications. Although the CIFAR10-C dataset was well suited to assess the CL algorithm's performance on natural corruptions and perturbations, it does not address the worst-case adversarial perturbations.

B. Adversarial robustness

An adversarial image is a clean image that was perturbed by a small change - carefully designed to confuse the classifier into making an erroneous decision. It has been demonstrated that neural networks are vulnerable to attacks on their input, which leads to incorrect classification [38]. In particular, most neural networks are highly vulnerable to attacks based on these small modifications of the input at test time [39], [40], [41].

Various algorithms have been developed to search for the smallest additive distortions in input space that confuse classifiers [42], [40], [43]. These algorithms create attacks (slightly perturbed input) that can be used to test (or exploit) the vulnerability of neural networks. Thus, adversarial distortions serve as a type of worst-case analysis for network robustness. Figures 1 and 5 present images from the clean CIFAR10 and adversarially created CIFAR10 datasets, respectively. The adversarially created dataset was created using the PGD and FGSM algorithms. The adversarial attack is stronger as the size of the perturbation (i.e, level of attack ϵ) increases. For example, in Figure 5, we can see that the dataset in row 1 was constructed with level of attack (ϵ) 1/255 and is easily accurately identified by the algorithm, as opposed to Row 17, with level of attack (ϵ) of 64/255, where the images are considerably more perturbed and distorted. To examine the stability of CL models trained on robust or non-robust features, in class incremental settings, we created 23 perturbed datasets using PGD-L_{inf}, PGD-L_2 and FGSM with varying level of attacks (ϵ), with min 1/255 and max of 224/225 [43]. As it has been shown in adversarial machine learning, the features of the input data play a prominent role in the robustness of the model [44]. Similarly, in CL learning, evaluating the five CL models against worst-case scenarios, the features of the input are also impacting the model's robustness and average accuracy; hence, they should be carefully selected (see the results and discussion Section for more information.)

IV. Experiments

A. Datasets

Three datasets were used to train all the models summarized in Table I. The first is standard CIFAR10($\mathcal{D}$), robust CIFAR10($\mathcal{R}$), and non-robust CIFAR10($\mathcal{N}$). Figures 1, 2, and 3 present examples for CIFAR10, robust CIFAR10, and non-robust CIFAR10, respectively. All three datasets are publicly available and can be found at CIFAR10[1], robust CIFAR10[2] and non-robust CIFAR10[3].

B. Creating adversarial datasets

We built adversarial examples using an independent VGG16 [45] model trained on the CIFAR10 dataset to compare the adversarial robustness of the five models. The adversarial datasets were created using PGD-L_{inf}, PGD-L_2 [43] and FGSM [40] algorithms for different values of the level of attacks (ϵ). The hyperparameter ϵ is defined as the maximum allowable perturbation to the original input in the optimization settings of PGD and FGSM attacks. In particular, ϵ determines the strength of the adversarial attack; the higher the value, the stronger the adversarial attack and the more perturbed the input image. The dataset created with PGD-L_{inf} can be seen in Figure 5.

C. Evaluation Metrics

We trained all five models in the class incremental learning (CIL) setting of continual learning. The first task consists of two classes, then one class was added sequentially for each following task. In total, we have nine tasks. We computed the average accuracy as follows:

$$ACC = \frac{1}{T} \sum_{i=1}^{T} R_{T,i}, \tag{4}$$

where R stands for the average accuracy, while the i stands for task index.

D. Protocols

Van *et al.* proposed three settings in continual learning [46]. In order of increasing complexity, they are task incremental learning (TIL), domain incremental learning (DIL), and class incremental learning (CIL). These settings allow for algorithms to be fairly compared to one another. We trained five models in the class incremental learning setting where groups of classes are sequentially observed. The CIL setting is significant as it assumes incoming data of new tasks and evaluates performance degradation of previous classes without a limit on the number of tasks; the CIL scenario is quite common in real-world situations.

E. Training

In total, five models were trained using three versions of the CIFAR-10 dataset: standard, robust and non-robust (see Figures 2 and 3). The average accuracies of the five models are outlined in Table I. The model $f_{(\mathcal{D},\mathcal{DD})}$ was trained on the standard CIFAR-10 dataset and the replay buffer contained only samples from the standard dataset. Model $f_{(\mathcal{D},\mathcal{DR})}$ was also trained on the standard CIFAR-10 dataset; however, the replay buffer contained an equal number of samples from the robust dataset along with the standard. Model $f_{(\mathcal{D},\mathcal{DN})}$ was also trained on the standard data set similar to Models $f_{(\mathcal{D},\mathcal{DD})}$ and $f_{(\mathcal{D},\mathcal{DR})}$; however, this model's replay buffer contains samples from the non-robust CIFAR-10 dataset along with the standard. Model $f_{(\mathcal{R},\mathcal{RD})}$ was trained on the robust CIFAR-10 dataset and its replay buffer consists of an equal number of samples from the standard and robust datasets. Model $f_{(\mathcal{N},\mathcal{ND})}$ was trained on the non-robust CIFAR-10 dataset and its replay buffer contained samples from the standard and the non-robust data sets. The replay buffer size (i.e, 16000) was kept equal during training of all models. The experiments were performed using the either Adam or SGD optimizer with learning rate of 0.01 or 0.1, respectively [47]. Random horizontal flipping was used as data augmentation during training for all five models. To make the comparison as fair as possible, we kept the number of epochs per task (i.e., 16), batch size (256), network architecture (i.e., WideResNet [48]), and momentum (i.e, 0.9) the same in the training of all five models. Each experiment was repeated five times with a different seed to achieve a better approximation. The optimum values of hyperparameters were determined using the grid search strategy.

V. Results and Discussion

Figure 6 shows the average validation accuracy of all five models outlined in Table I. Model $f_{(\mathcal{D},\mathcal{DD})}$ obtained the highest accuracy on clean data followed by model $f_{(\mathcal{D},\mathcal{DR})}$ and then model $f_{(\mathcal{D},\mathcal{DN})}$. These three models were trained on the same standard dataset but differ in the replay buffers. Model $f_{(\mathcal{N},\mathcal{ND})}$ had the lowest accuracy. Models $f_{(\mathcal{R},\mathcal{RD})}$ and $f_{(\mathcal{N},\mathcal{ND})}$ were not trained on the standard datasets. However, when models $f_{(\mathcal{R},\mathcal{RD})}$ and $f_{(\mathcal{N},\mathcal{ND})}$ completed learning all nine tasks, their average accuracy was comparable to models $f_{(\mathcal{D},\mathcal{DD})}$, $f_{(\mathcal{D},\mathcal{DR})}$ and $f_{(\mathcal{D},\mathcal{DN})}$. Interestingly, model $f_{(\mathcal{N},\mathcal{ND})}$ has the lowest accuracy on the first task but as it learned more tasks, it progressively increased its average accuracy. Its average accuracy was comparable to models $f_{(\mathcal{D},\mathcal{DD})}$, $f_{(\mathcal{D},\mathcal{DR})}$ and $f_{(\mathcal{D},\mathcal{DN})}$ at the end of learning task nine, even though it was trained solely on non-robust features. The unique behavior of model $f_{(\mathcal{N},\mathcal{ND})}$ suggests that it is accumulating transferable knowledge during the sequential learning of these tasks. Model $f_{(\mathcal{N},\mathcal{ND})}$'s average accuracy on task one was approximately 45%. After learning task two, its average accuracy jumped to approximately 52% (as opposed to the decreasing accuracy observed in the other four models). This phenomenon is called *backward transfer*. This empirically demonstrates that non-robust features carry transferable knowledge across tasks. The

[1] CIFAR10: https://www.cs.toronto.edu/ kriz/cifar.html

[2] Robust CIFAR10: https://github.com/MadryLab/constructed-datasets

[3] Non-robust CIFAR10: https://github.com/MadryLab/constructed-datasets

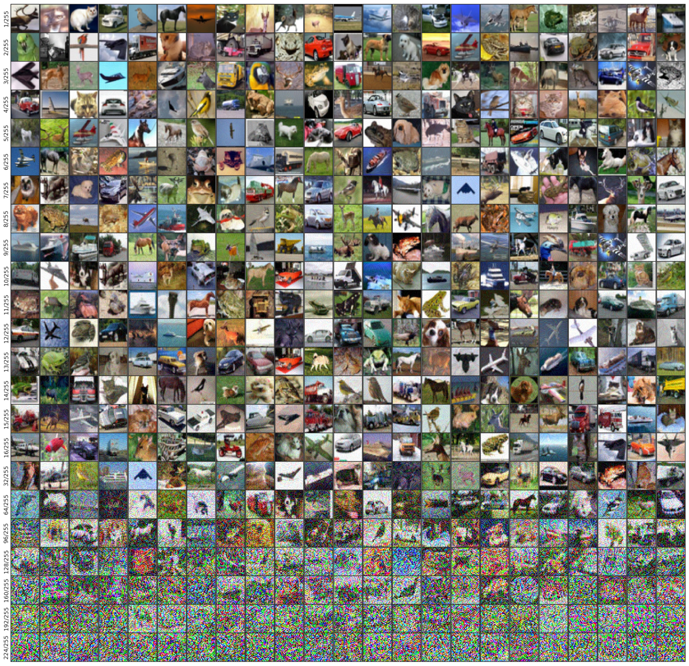

Fig. 5. Adversarial examples created using PGD-L_{inf}. On the x-axis, we can see the different adversarially perturbed images from the dataset. The different strengths of the level of the attacks (ϵ) are plotted in the y-axis. The level of the attack (ϵ) determines the strength of the adversarial attack; the higher the value, the stronger the adversarial attack and the more perturbed the input image (Best viewed in color).

TABLE I
THE AVERAGE ACCURACIES OF ALL 5 MODELS. $\mathcal{D}$ = STANDARD CIFAR10, $\mathcal{R}$ = ROBUSTIFIED CIFAR10 AND $\mathcal{N}$ = NON-ROBUSTIFIED CIFAR10. IN MODEL $f_{(X,YZ)}$: THE FIRST ENTRY REPRESENTS THE TRAINING DATA SET, THE SECOND SET OF LETTERS DENOTES THE REPLAY BUFFER DATASETS SAMPLED EQUALLY.

Model	**Training set**	**Replay buffer** ($size$ = 16000)	**Average accuracy**
$f_{(\mathcal{D},\mathcal{D}\mathcal{D})}$	CIFAR10 ($\mathcal{D}$)	CIFAR10 ($\mathcal{D}$) + CIFAR10 ($\mathcal{D}$)	$88.20{\pm}_{0.48}$
$f_{(\mathcal{D},\mathcal{D}\mathcal{R})}$	CIFAR10 ($\mathcal{D}$)	CIFAR10 ($\mathcal{D}$) + Robustified CIFAR10 ($\mathcal{R}$)	$85.99{\pm}_{0.77}$
$f_{(\mathcal{D},\mathcal{D}\mathcal{N})}$	CIFAR10 ($\mathcal{D}$)	CIFAR10 ($\mathcal{D}$) + Non-Robustified CIFAR10 ($\mathcal{N}$)	$82.72{\pm}_{0.49}$
$f_{(\mathcal{R},\mathcal{R}\mathcal{D})}$	Robustified CIFAR10 ($\mathcal{R}$)	Robustified CIFAR10 ($\mathcal{R}$) + CIFAR10 ($\mathcal{D}$)	$83.12{\pm}_{1.9}$
$f_{(\mathcal{N},\mathcal{N}\mathcal{D})}$	Non-Robustified CIFAR10 ($\mathcal{N}$)	Non-Robustified CIFAR10 ($\mathcal{N}$) + CIFAR10 ($\mathcal{D}$)	$80.40{\pm}_{1.6}$

model will be able to acquire more generalizable knowledge among tasks as the task sequence grows larger.

Similarly, Model $f_{(\mathcal{D},\mathcal{D}\mathcal{N})}$ contained non-robust features in its replay buffer. It shows a similar overall trend to model $f_{(\mathcal{N},\mathcal{N}\mathcal{D})}$ in that as it learns more tasks it is able to increase its average accuracy. We do see a dip reaching up to task five, which can be attributed to the small sample size of non-robust features contained in the replay buffer, so there is not as much transferable knowledge that the model is able to collect. As the model learns more than five tasks, it increases its accuracy.

These experiments were based on learning nine tasks. However, given the trends shown in Figure 6, we can hypothesize that if the number of tasks was significantly greater than nine, then a model trained solely on non-robust features could continuously build its transferable knowledge and perform the same, or better, than the standard model (Model $f_{(\mathcal{D},\mathcal{D}\mathcal{D})}$).

Model $f_{(\mathcal{R},\mathcal{R}\mathcal{D})}$ performed equal to the highest performing model on task one. However, as it learned the second task, there was a steep decrease in its average accuracy. This is because there is not much transferable knowledge in robust features; they are more class-specific and sparse as compared to non-robust features. Of note, model $f_{(\mathcal{R},\mathcal{R}\mathcal{D})}$ was still able to learn using robust features while maintaining a higher overall accuracy on all tasks compared to the model trained on non-robust features (model $f_{(\mathcal{N},\mathcal{N}\mathcal{D})}$).

A. Evaluating robustness against common corruptions

All five models were trained on varying datasets and replay buffers, as mentioned in Table I. In Figure 7, we tested these five models against the 19 types of noise. The x-axis of each graph shows the severity level, ranging from 0-5, 0 being a clean image and 5 being the most severe corruption. The y-axis displays the average accuracy. The model trained on robust features, $f_{(\mathcal{R},\mathcal{R}\mathcal{D})}$, obtained the highest average accuracy in 12

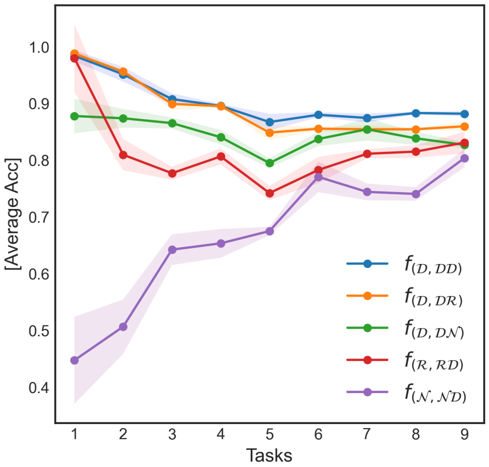

Fig. 6. The average accuracy of the 5 models $f_{(\mathcal{D},\mathcal{DD})}$, $f_{(\mathcal{D},\mathcal{DR})}$, $f_{(\mathcal{D},\mathcal{DN})}$, $f_{(\mathcal{R},\mathcal{RN})}$ and $f_{(\mathcal{N},\mathcal{ND})}$ as they incrementally learn a sequence of 9 tasks in CIPHAR10. Where $\mathcal{D}$ = Standard CIFAR10, $\mathcal{R}$ = Robustified CIFAR10 and $\mathcal{N}$ = Non-robustified CIFAR10 datasets (Best viewed in color).

out of the 19 types of noise. The non-robust models $f_{(\mathcal{D},\mathcal{DN})}$ and $f_{(\mathcal{N},\mathcal{ND})}$ performed the worst in all 19 types of noise.

As the severity of the noise increases, the robust model shows its strength by maintaining its average accuracy. Model $f_{(\mathcal{D},\mathcal{DR})}$ had lower average accuracies than model $f_{(\mathcal{D},\mathcal{DD})}$ but performed better than the other three models (i.e, $f_{(\mathcal{D},\mathcal{DD})}$, $f_{(\mathcal{N},\mathcal{ND})}$ and $f_{(\mathcal{D},\mathcal{DN})}$) because its replay buffer contained robust features.

Model $f_{(\mathcal{D},\mathcal{DD})}$ - the standard model - performed better than the non-robust models ($f_{(\mathcal{D},\mathcal{DN})}$ and $f_{(\mathcal{N},\mathcal{ND})}$) under all 19 forms of noise. However, it performed worse than the robust models ($f_{(\mathcal{D},\mathcal{DR})}$ and $f_{(\mathcal{R},\mathcal{RD})}$) on 12 noise types.

B. Evaluating robustness against adversarial examples

Figure 8, 9 and 10 show the average accuracies of the five models under adversarial examples generated using PGD-L_{inf}, PGD-L_2 and FGSM algorithms. Observe that model $f_{(\mathcal{R},\mathcal{RD})}$, trained on robust features, performed the best against adversarial perturbations compared to all other models. The second best performing model was $f_{(\mathcal{D},\mathcal{DR})}$. That is, the models that contained some robust features, whether in the training set or the replay buffer performed the best. The models that performed the worst trained on non-robust features. Model $f_{(\mathcal{D},\mathcal{DD})}$ was right in between the four models. As we work on training models in continual learning, it is vital that we pay attention to the feature selection involved during the training process as it directly influences the strength of the model against perturbations. As shown in Figure 6, models can be trained on any type of features; however, it is important to note that inclusion of robust features results in more robust models.

VI. Conclusion

Recent approaches in continual learning focus on extracting various types of features from multi-task datasets to prevent catastrophic forgetting — without formally evaluating the quality, robustness and usefulness of these features. In this paper, we presented an empirical and exhaustive study to demonstrate the crucial role of features in the context of class incremental learning (CIL) under various noise and perturbation environments. Our experiments demonstrate that a set of models trained on the standard (a mixture of both robust and non-robust) features obtained a higher accuracy on clean data compared to the models trained either using robust features or non-robust features. However, the models trained using standard and non-robust features performed poorly in noisy and adversarial conditions as compared to the model trained using robust features. The model trained using non-robust features performed the worst in noisy conditions and under adversarial attacks. Our study underlines the significance of using robust features in CIL. Our code is available at https://github.com/hikmatkhan/ARCL

References

[1] M. B. Ring, "Continual learning in reinforcement environments," *PhD thesis, University of Texas at Austin*, 1994.

[2] S. Thrun and T. M. Mitchell, "Lifelong robot learning," *Robotics and autonomous systems*, vol. 15, no. 1-2, pp. 25–46, 1995.

[3] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," in *Psychology of learning and motivation*. Elsevier, 1989, vol. 24, pp. 109–165.

[4] R. Ratcliff, "Connectionist models of recognition memory: constraints imposed by learning and forgetting functions." *Psychological review*, vol. 97, no. 2, p. 285, 1990.

[5] M. Mermillod, A. Bugaiska, and P. Bonin, "The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects," *Frontiers in psychology*, vol. 4, p. 504, 2013.

[6] H. Qu, H. Rahmani, L. Xu, B. Williams, and J. Liu, "Recent advances of continual learning in computer vision: An overview," *arXiv preprint arXiv:2109.11369*, 2021.

[7] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 12, pp. 2935–2947, 2017.

[8] P. Dhar, R. V. Singh, K.-C. Peng, Z. Wu, and R. Chellappa, "Learning without memorizing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5138–5146.

[9] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 114, no. 13, pp. 3521–3526, 2017.

[10] F. Zenke, B. Poole, and S. Ganguli, "Continual learning through synaptic intelligence." *Proceedings of machine learning research*, vol. 70, pp. 3987–3995, 2017.

[11] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars, "Memory aware synapses: Learning what (not) to forget," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 139–154.

[12] G. Saha, I. Garg, and K. Roy, "Gradient projection memory for continual learning," in *International Conference on Learning Representations (ICLR)*, 2020.

[13] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "icarl: Incremental classifier and representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2001–2010.

[14] A. Robins, "Catastrophic forgetting, rehearsal and pseudorehearsal," *Connection Science*, vol. 7, no. 2, pp. 123–146, 1995.

[15] M. Riemer, I. Cases, R. Ajemian, M. Liu, I. Rish, Y. Tu, , and G. Tesauro, "Learning to learn without forgetting by maximizing transfer and minimizing interference," in *International Conference on Learning Representations (ICLR)*, 2019.

[16] D. Lopez-Paz and M. Ranzato, "Gradient episodic memory for continual learning," *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017.

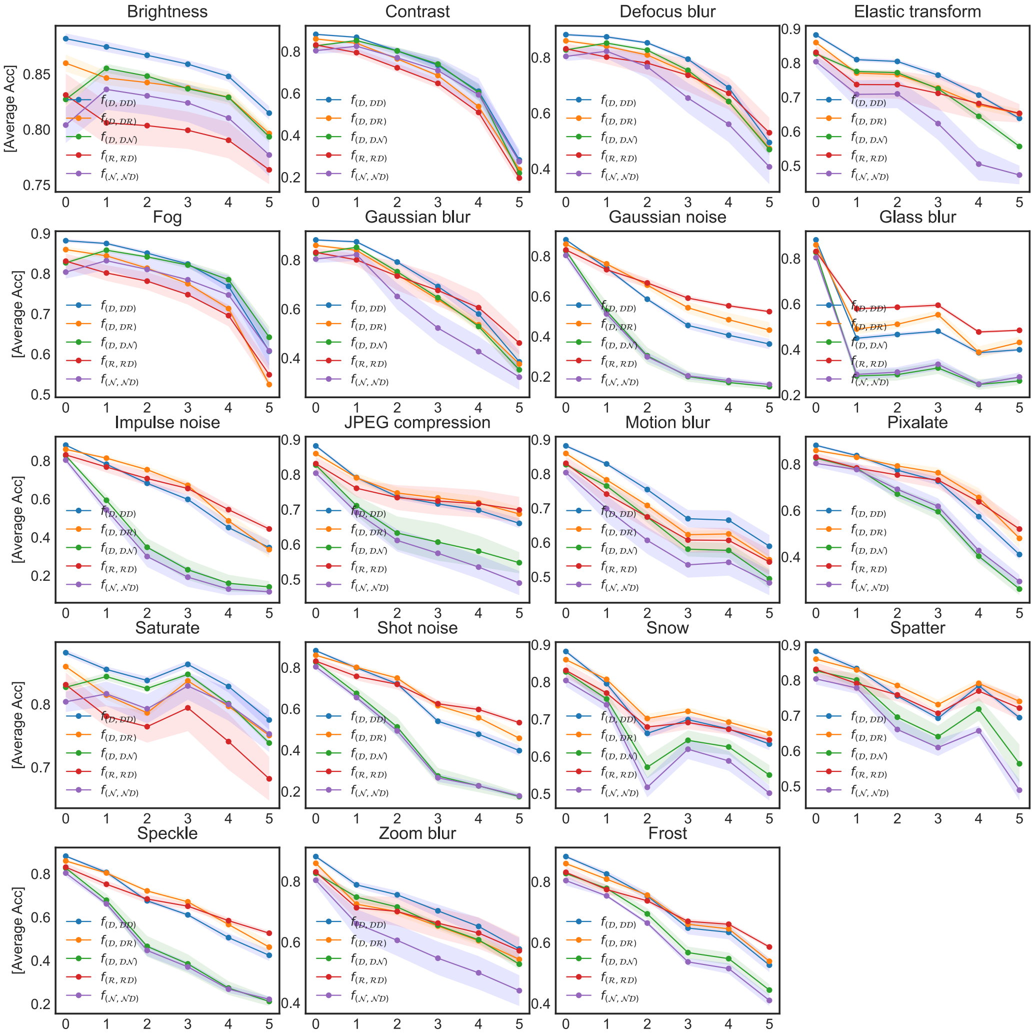

Fig. 7. The average accuracy of the 5 models (i.e, $f_{(\mathcal{D},\mathcal{DD})}$, $f_{(\mathcal{D},\mathcal{DR})}$, $f_{(\mathcal{D},\mathcal{DN})}$, $f_{(\mathcal{R},\mathcal{RN})}$ and $f_{(\mathcal{N},\mathcal{ND})}$) against each of the 19 types of noise. The y-axis is the average accuracy. The x-axis is the severity of the noise, 5 being the most severe. $\mathcal{D}$ = Standard CIFAR10, $\mathcal{R}$ = Robustified CIFAR10 and $\mathcal{N}$ = Non-robustified CIFAR10 datasets. All models evaluated five times (Best viewed in color).

[17] A. Chaudhry, N. Khan, P. Dokania, and P. Torr, "Continual learning in low-rank orthogonal subspaces," *Advances in Neural Information Processing Systems (NIPS)*, vol. 33, pp. 9900–9911, 2020.

[18] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," *Advances in Neural Information Processing Systems (NIPS)*, vol. 30, 2017.

[19] N. Kamra, U. Gupta, and Y. Liu, "Deep generative dual memory network for continual learning," *arXiv preprint arXiv:1710.10368*, 2017.

[20] G. M. van de Ven, H. T. Siegelmann, and A. S. Tolias, "Brain-inspired replay for continual learning with artificial neural networks," *Nature communications*, vol. 11, no. 1, pp. 1–14, 2020.

[21] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra, "Pathnet: Evolution channels gradient descent in super neural networks," *arXiv preprint arXiv:1701.08734*, 2017.

[22] J. Yoon, E. Yang, J. Lee, and S. J. Hwang, "Lifelong learning with dynamically expandable networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[23] S. Ebrahimi, F. Meier, R. Calandra, T. Darrell, and M. Rohrbach, "Adversarial continual learning," in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020, pp. 386–402.

[24] A. Mallya and S. Lazebnik, "Packnet: Adding multiple tasks to a single network by iterative pruning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7765–7773.

[25] P. Kaushik, A. Kortylewski, A. Gain, and A. Yuille, "Understanding catastrophic forgetting and remembering in continual learning with optimal relevance mapping," in *Fifth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*, 2021.

[26] G. Saha and K. Roy, "Saliency guided experience packing for replay in continual learning," *arXiv preprint arXiv:2109.04954*, 2021.

[27] S. Ebrahimi, S. Petryk, A. Gokul, W. Gan, J. E. Gonzalez, M. Rohrbach, and T. Darrell, "Remembering for the right reasons: Explanations reduce catastrophic forgetting," *Applied AI Letters*, vol. 2, no. 4, p. e44, 2021.

[28] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[29] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2018, pp. 839–847.

[30] K. Hikmat, R. Ghulam, B. Nidhal, C, , T. Tyler, T. Lacey, and J. Charles C, "Explainable ai: Rotorcraft attitude prediction," *Vertical Flight Society's 76th Annual Forum Technology Display, Virginia Beach, Virginia, USA*, Oct 2020.

[31] K. Hikmat, R. Ghulam, B. Nidhal, C, and J. Charles C, "Rotorcraft flight information inference from cockpit videos using deep learning," *Ameri-*

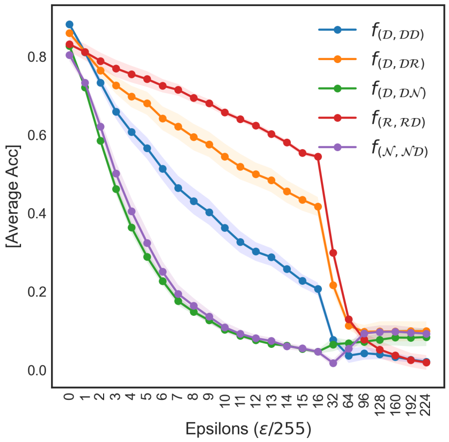

Fig. 8. The average accuracy of five models (i.e, $f_{(\mathcal{D},\mathcal{DD})}$, $f_{(\mathcal{D},\mathcal{DR})}$, $f_{(\mathcal{D},\mathcal{DN})}$, $f_{(\mathcal{R},\mathcal{RN})}$ and $f_{(\mathcal{N},\mathcal{ND})}$) against the strength of the PGD-L_{inf} attack. The x-axis displays different strengths of the level of attacks (ϵ/255), and the y-axis displays the average accuracy. We observe that the model ($f_{(\mathcal{N},\mathcal{ND})}$) trained on robust CIFAR10 ($\mathcal{D}_{\mathcal{R}}$) maintained its average accuracy while models that were either trained or had non-robust CIFAR10 (i.e., samples from either $f_{(\mathcal{N}or\mathcal{D})}$) samples in their replay buffer performed the worst. $\mathcal{D}$ = Standard CIFAR10, $\mathcal{R}$ = Robustified CIFAR10 and $\mathcal{N}$ = Non-robustified CIFAR10 datasets (Best viewed in color).

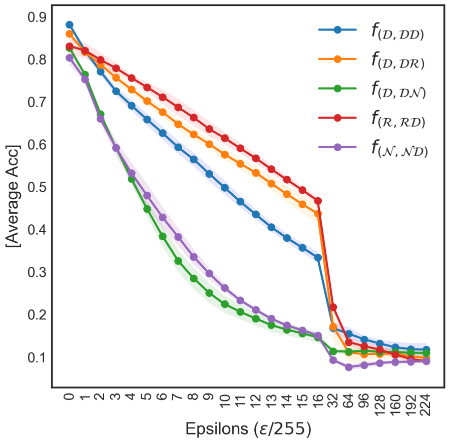

Fig. 9. The average accuracy of five models (i.e, $f_{(\mathcal{D},\mathcal{DD})}$, $f_{(\mathcal{D},\mathcal{DR})}$, $f_{(\mathcal{D},\mathcal{DN})}$, $f_{(\mathcal{R},\mathcal{RN})}$ and $f_{(\mathcal{N},\mathcal{ND})}$) against the strength of the FGSM attack. The x-axis displays different strengths of the level of attack (ϵ), and the y-axis displays the average accuracy. We observe that the model ($f_{(\mathcal{N},\mathcal{ND})}$) trained on robust CIFAR10 ($\mathcal{D}_{\mathcal{R}}$) maintains its average accuracy while models that were either trained or had non-robust CIFAR10 (i.e., samples from either $f_{(\mathcal{N}or\mathcal{D})}$) samples in their replay buffer performed the worst. $\mathcal{D}$ = Standard CIFAR10, $\mathcal{R}$ = Robustified CIFAR10 and $\mathcal{N}$ = Non-robustified CIFAR10 datasets (Best viewed in color).

can Helicopter Society 75th Annual Forum, Philadelphia, Pennsylvania, USA, May 2019.

[32] K. Hikmat, R. Ghulam, B. Nidhal, C, , T. Tyler, T. Lacey, and J. Charles C, "Deep ensemble for rotorcraft attitude prediction," *Vertical Flight Society's 77th Annual Forum Technology Display, Palm Beach, Florida, USA*, May 2021.

[33] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International Conference on Machine Learning (ICML)*. PMLR, 2014, pp. 647–655.

[34] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems (NIPS)*, vol. 27, 2014.

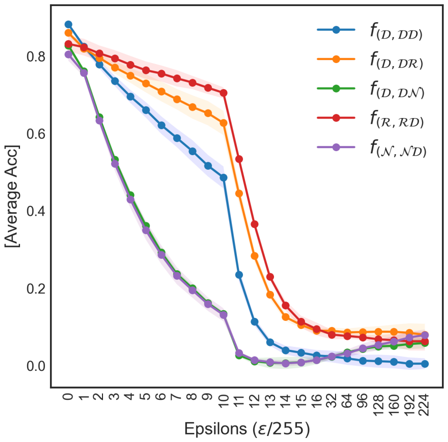

Fig. 10. The average accuracy of five models (i.e, $f_{(\mathcal{D},\mathcal{DD})}$, $f_{(\mathcal{D},\mathcal{DR})}$, $f_{(\mathcal{D},\mathcal{DN})}$, $f_{(\mathcal{R},\mathcal{RN})}$ and $f_{(\mathcal{N},\mathcal{ND})}$) against the strength of the PGD-L_2 attack. The x-axis displays values for , and the y-axis displays the average accuracy. We observe that the model ($f_{(\mathcal{N},\mathcal{ND})}$) trained on robust CIFAR10 ($\mathcal{D}_{\mathcal{R}}$) maintained its average accuracy while models that were either trained or had non-robust CIFAR10 (i.e., samples from either $f_{(\mathcal{N}or\mathcal{D})}$) samples in their replay buffer performed the worst. $\mathcal{D}$ = Standard CIFAR10, $\mathcal{R}$ = Robustified CIFAR10 and $\mathcal{N}$ = Non-robustified CIFAR10 datasets (Best viewed in color).

[35] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in Neural Information Processing Systems (NIPS)*, vol. 32, 2019.

[36] Y.-Y. Yang, C. Rashtchian, H. Zhang, R. R. Salakhutdinov, and K. Chaudhuri, "A closer look at accuracy vs. robustness," *Advances in Neural Information Processing Systems (NIPS)*, vol. 33, pp. 8588–8601, 2020.

[37] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *International Conference on Learning Representations (ICLR)*, 2018.

[38] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *2nd International Conference on Learning Representations (ICLR)*, 2014.

[39] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Proceedings of the 2013th European Conference on Machine Learning and Knowledge Discovery in Databases-Volume Part III*, 2013, pp. 387–402.

[40] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2015.

[41] N. Papernot, P. Mcdaniel, and I. J. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *ArXiv*, vol. abs/1605.07277, 2016.

[42] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, 2017, pp. 39–57.

[43] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations (ICLR)*, 2018.

[44] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *International Conference on Learning Representations (ICLR)*, 2018.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2015.

[46] G. M. Van de Ven and A. S. Tolias, "Three scenarios for continual learning," *Advances in Neural Information Processing Systems (NIPS)*, 2019.

[47] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015.

[48] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.