Towards Generalizable Robustness of Deep Learning Models

Muhammad Ahmed Shah

CMU-LTI-25-009

March 2025

Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Bhiksha Raj (Chair)
Rita Singh
Barbara Shinn-Cunningham
Richard Stern
Josh McDermott (MIT)

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy In Language and Information Technologies

Abstract

While Deep Neural Networks (DNNs) have advanced the state-of-the-art in machine perception by leaps and bounds, their sensitivity to subtle input perturbations that humans are invariant to has raised questions about their reliability in real-world settings. Perhaps the most pernicious and alarming of these perturbations are *adversarial perturbations*, which can drastically and arbitrarily change the outputs of DNNs, while remaining imperceptible (or barely perceptible) to humans. Algorithms for generating adversarial perturbations are known as adversarial attacks.

Existing methods of making DNNs robust to adversarial perturbations, generally require training on adversarially perturbed or noisy data. While this approach successfully produces DNNs robust to the adversarial attacks used to generate perturbations during training, it does not generalize to other, unseen, types of attacks. Consequently, to obtain models that exhibit a more *generalized robustness* to a variety of adversarial attacks one would need to ensure that all such attacks are sufficiently represented in the training data. This objective is highly inefficient at best, or impossible, at worst, given that adversarial attacks are constantly evolving and the boundaries of human perception are not fully known.

Given the pitfalls of seeking robustness via training, in this thesis, we work towards models that are *naturally* more robust to a variety of adversarial attacks *without having been trained on perturbed data*. To this end, we seek to discover principles, or *priors*, for DNNs that endow them with enhanced robustness to adversarial perturbations. As these priors induce adversarial robustness without requiring training on perturbed data, we expect them to yield models robust to various perturbations and attack algorithms.

Concretely, we study two categories of robustness priors in this thesis: structure and biological. We define structural robustness priors as design elements of DNNs that are conducive to adversarial robustness. Biological priors, on the other hand, are mechanisms and constraints related to the robustness of biological perception and cognition but are not usually represented in DNNs. Since adversarial perturbations are rooted in the difference between biological perception and DNNs, we expect that integrating biological priors into DNNs would better align their behavior with biological perception and consequently cause them to exhibit robustness to adversarial perturbations, and perhaps even various other noises that biological perception is robust to.

We approach the study of structural robustness priors from two directions, namely statistical, and empirical. In the former, we take the view that by virtue of being highly overparameterized modern DNNs may encode spurious features, and show that pruning away neurons that encode such spurious features improves robustness to adversarial attacks. In the empirical approach, we estimate the probability with which gradient descent, from a random initialization, arrives at a model that is both robust and accurate. Our experiments on simple problems, like XOR or MNIST, reveal that certain design elements increase the odds of finding robust models while others decrease these odds.

In our study of biological priors, we consider sensory and cognitive priors. Sensory priors relate to the constraints present in sensor organs that emphasize or de-emphasize certain aspects of the stimuli. In the domain of vision, one such prior is foveation due to which only the region around the fixation point is sensed at maximum fidelity. We integrate foveation in DNNs and demonstrate

that it significantly improves their robustness to adversarial attacks, as well as non-adversarial perturbations. Similarly, examples biological priors in audition are simultaneous frequency masking and lateral suppression due to which the perceived level of a frequency is influenced by the levels of other adjacent frequencies. We integrate these phenomena into speech recognition DNNs and observe that their robustness to adversarial attacks, as well as other corruptions, is greatly enhanced while their accuracy is minimally impacted.

Cognitive priors, on the other hand, relate to the computations performed in the brain. In this connection, we have explored the role of inflexible inter-neuron correlations and shown that constraining the inter-neuron correlations makes DNNs more robust to adversarial and non-adversarial perturbations. We have also simulated feedback connections, that are ubiquitous in the brain, in DNNs and shown that doing so improves adversarial robustness.

To reliably evaluate the improvements we achieve and compare them with prior work, we need standardized robustness benchmarks. While such benchmarks have been developed for vision tasks, they do not exist for other modalities such as audio. To fill this gap, we have developed a comprehensive robustness benchmark for speech models called Speech Robust Bench (SRB). SRB is composed of 114 challenging speech recognition scenarios covering the range of corruptions that ASR models may encounter when deployed in the wild.

Contents

Chapter 1

Introduction

1.1 Motivation

Deep Neural Networks (DNNs) are exceptionally adept at many computer vision tasks and have emerged as one of the best models of the biological neurons involved in visual object recognition [Yamins et al., 2014, Cadieu et al., 2014]. However, their lack of robustness to subtle image perturbations that humans are largely invariant to Szegedy et al. [2014], Geirhos et al. [2018b], Dodge and Karam [2017] has raised questions about their reliability in real-world scenarios. Of these perturbations, perhaps the most pernicious are *adversarial perturbations*, which are specially crafted distortions that can change the response of DNNs when added to their inputs [Szegedy et al., 2014, Ilyas et al., 2019] but are either imperceptible to humans or perceptually irrelevant enough to be ignored by them. Algorithms for generating adversarial perturbations are known as adversarial attacks.

The dominant approach for making DNNs robust to adversarial attacks involves exposing them to adversarially perturbed images [Madry et al., 2018b, Wong et al., 2019a, Zhang et al., 2019] (adversarial training) or random noise [Cohen et al., 2019a, Fischer et al., 2020, Carlini et al., 2022] during training. While this approach is highly effective in making DNNs robust to the types of adversarial attacks used during training, the robustness often does not generalize to other, unseen, types of attacks. [Joos et al., 2022, Sharma and Chen, 2017, Schott et al., 2018]. For example, given that most adversarial attacks generate ℓ_p norm bounded perturbations, the complementary regions between the norm balls for different values of p leave allow adversaries operating on a different p than one used during training to successfully attack the DNN [Joos et al., 2022]. Furthermore, there are other classes of adversarial attacks, such as patch attacks [Xu et al., 2023, Sharma et al., 2022], transformation-based attacks [Xiao et al., 2018, Kang et al., 2019, Laidlaw and Feizi, 2019], and attacks that exploit the null spaces of human perception [Qin et al., 2019, Laidlaw et al., 2021], that are not subsumed by ℓ_p norm bounds, and against which DNNs trained with ℓ_p bounded perturbations confer limited robustness [Laidlaw et al., 2021, Hsiung et al., 2023]. Consequently, under this paradigm of robustness via training, to obtain a model with *generalized robustness* to all the myriad types of adversarial attacks one would need to simulate them all during training, a task which may prove to be prohibitively expensive. Given the lack of generalizability, even if one were able to do that, the robustness achieved by the model would likely be limited to the existing adversarial attacks, and the extent to which it would generalize to other attacks would

remain uncertain. Given this context, we present our thesis statement below.

1.2 Thesis Statement

The goal of this thesis is to discover principles, or *priors*, that enhance the adversarial robustness of DNNs *without explicitly training on adversarially perturbed data*. Particularly, we consider two types of priors: (1) Structural priors and, (2) biological priors.

Structural priors are the design elements of DNNs that are conducive to adversarial robustness. These can include the width and depth of the DNN, the choice of activation and normalization functions, and regularization methods. Since prior work demonstrates that by training on perturbed data, the same DNN can learn an adversarial robust boundary, we hypothesize that perhaps modifications to certain structural elements may bias the model towards such a boundary without explicitly training.

Biological priors, on the other hand, are mechanisms and constraints considered to contribute to the robustness of biological perception. Since adversarial attacks, by definition, exist due to differences between biological perception and DNNs, we expect that integrating biological priors into DNNs would make them better aligned with biological perception, which, in turn, would make them more robust to adversarial perturbations.

In order that we may accurately evaluate the progress we make and compare our methods with prior work, we also develop standardized robustness benchmarks. Specifically, we address the lack of a standardized robustness benchmark for speech tasks by developing one that evaluates robustness to a variety of transforms, including adversarial attacks.

In a word, in this thesis, we identify several structural and biological priors that make DNNs trained, only on natural data, more robust to a variety of adversarial attacks, and standardized robustness benchmark for speech DNNs. Further details about the priors we identified are presented in the subsequent sections.

1.3 Structural Priors

As mentioned above, structural priors are the design elements of DNNs that are conducive to adversarial robustness. In this thesis, we approach the study of structural robustness priors from the following two directions. In the first approach, we take the view that adversarial attacks exist because DNNs are sensitive to spurious features, such as high-frequency components Wang et al. [2020] and texture Geirhos et al. [2018a], and we seek to reduce their sensitivity to these features by pruning away substructures within the network that encode these features. In the second approach, we empirically determine the extent to which various DNN design choices, such as width, depth, activation function, and regularization, bias the model toward learning robust decision boundaries. To this end, we estimate the probability with which gradient descent, starting from random initialization, can converge to a solution corresponding to an adversarially robust decision boundary. We present further details about each of these approaches below.

1.3.1 Pruning Spurious Substructures for Robustness

It has been observed that DNNs utilize spurious features to make their predictions Ilyas et al. [2019]. Examples of such spurious features are high-frequency components Wang et al. [2020] and texture Geirhos et al. [2018a]. These features are somewhat correlated with the true labels, either coincidentally or due to sampling biases in the training data, however, they are often meaningless for humans Wang et al. [2020]. Therefore an adversary can craft adversarial perturbations that modify these spurious features and induce miscalssifications in the DNN, while remaining imperceptible or irrelevant to humans Geirhos et al. [2018a]. We seek to mitigate the DNN's reliance on spurious features by pruning away substructures of DNNs that encode them. In this connection, we leverage the interpretation of neurons in a DNN as being feature detectors, to draw an equivalence between neurons and the features themselves. Since removing the detector for a feature effectively removes the influence of that feature from the DNN, if we identify and prune away neurons that encode spurious features, we will have effectively removed the influence of the spurious features from the DNN. To this end we develop a technique that identifies spurious neurons using various metrics, such as colinearilty, the magnitude of derivative (w.r.t the loss), and mutual information with the true label, and prunes them away. We demonstrate that post-pruning the DNNs become more robust to adversarial attacks, even surpassing adversarial training in cases when the test time adversarial perturbations were larger than the ones during training.

1.3.2 Influence of DNN Design on Odds of Finding Robust Boundary

The fact that a DNN exhibits almost no robustness to adversarial attacks when trained on clean data but becomes highly robust when it is trained on adversarially perturbed data indicates that it is capable of learning both robust and non-robust decision boundaries. We hypothesize that similar to how training on adversarially perturbed data biases the optimization towards robust solutions, certain structural elements of the DNN may do the same, and serve as structural robustness priors. We develop a framework to identify these structural robustness priors by modifying the structural elements of a DNN, and approximating the probability that gradient descent, starting from random initialization, can converge to a solution corresponding to an adversarially robust decision boundary. Our experiments on simple problems, like XOR or MNIST, reveal that certain design elements, like larger width and dropout, increase the odds of finding robust decision boundaries while others, like larger depth and batch normalization, decrease these odds.

1.4 Biological Priors

As mentioned above, we consider biological priors to be mechanisms and constraints considered to contribute to the robustness of biological perception. Adversarial attacks, by definition, exist due to differences between biological perception and DNNs, because they sample the space of perturbations that human perception is invariant to but DNNs are highly sensitive to. Therefore, we posit that integrating biological priors into DNNs would make them better aligned with biological perception, thereby reducing the space of adversarial perturbations and thus making the DNNs more robust to adversarial perturbations. Since human perception is robust to a wide variety of perturbations that DNNs are often confuse by, we expect that integrating biological priors

into DNNs would make them robust to various other noises (apart from adversarial attacks) that human perception is robust to. Indeed there is evidence indicating a positive correlation between biological alignment and adversarial robustness Dapello et al. [2020], Harrington and Deza [2021]. Moreover, a small but growing body of work Paiton et al. [2020], Bai et al. [2021], Dapello et al. [2020], Jonnalagadda et al. [2022], Luo et al. [2015], Gant et al. [2021], Vuyyuru et al. [2020] has shown that integrating biological mechanisms into DNNs improves their robustness to adversarial attacks, as well as non adversarial perturbations. In this thesis we build upon this body of work by considering biological priors that have not been studied as yet. Specifically, we consider two types of biological priors, namely sensory and cognitive.

1.4.1 Sensory Priors

We define sensory priors as the constraints imposed by the biological sensory organs that emphasize and de-emphasize certain characteristics of the incoming stimuli. In the domain of vision, we study foveation as sensory robustness prior. Foveation is the phenomenon that causes only the central 1% of the visual field to be sensed with maximum fidelity [Kolb, 2005, Stewart et al., 2020], while the rest of it lacks sharpness and color saturations Hansen et al. [2009]. This is unlike DNN, which view the entire image in full fidelity. We hypothesize that the experience of viewing the world at multiple levels of fidelity, perhaps even at the same instant, causes human vision to be invariant to low-level features, such as textures, and high-frequency patterns, that can be exploited by adversarial attacks. To test this hypothesis, we develop *R-Blur* (short for Retina Blur), which simulates foveation by blurring the image and reducing its color saturation adaptively based on the distance from a given fixation point. We find that models augmented with *R-Blur* retain most of the high classification accuracy of the base ResNet while being more robust to both adversarial and non-adversarial image corruptions, and that the adversarial robustness achieved by *R-Blur* is certifiable using the approach from Cohen et al. [2019a].

We extend our work on sensory priors to the auditory domain by studying the impact of biologically plausible feature extraction methods on the adversarial robustness of speech processing models. While there is significant literature on biologically derived acoustic features [Feather et al., 2019, Kim and Stern, 2016], modern speech processing systems generally operate on either the raw waveform [Baevski et al., 2020] or the log mel spectrogram [Radford et al., 2023]. While the use of the Mel filterbank does impart a degree of biological plausibility to the log Mel spectrogram, it does not represent several processes that occur in the cochlea and the auditory nerve, including filtering, more realistic non-linearity, lateral suppression and cross/auto-correlation [Stern and Morgan, 2012]. Adversarial attacks are known to exploit these discrepancies between DNNs and biological perception to generate perturbations that compromise DNNs but remain imperceptible. For example, lateral suppression induces simultaneous frequency masking in humans Stern and Morgan [2012] whereby the perceptual thresholds of the frequencies near a loud frequency are increased. However, since frequency masking does not occur in DNNs the attack proposed by Qin et al. [2019] was able to add fairly large magnitude noise in frequencies adjacent to loud frequencies and change the responses of speech recognition models, while remaining largely undetectable by humans. We investigate the impact, vis-a-vis robustness to adversarial and non-adversarial perturbations, of various biologically derived audio/speech feature extraction methods. In addition to evaluating existing features like log-Mel features, cochleagrams and Power Normalized Cepstral Coefficients (PNCC) Kim and Stern [2016], we have developed and evaluated novel features that

simulate simultaneous frequency masking and lateral suppression. We find that log Mel features, despite being the feature of choice in state-of-the-art speech recognition models, is among the least robust features we evaluated. We also observe that our proposed features are highly robust to adversarial attacks and non-adversarial noise, while having accuracy similar to or better than log Mel features.

1.4.2 Cognitive Priors

We define cognitive priors as mechanisms and constraints that influence how the sensory information is processed to generate perception. In this connection, we simulate the phenomenon of inflexible inter-neuron correlations observed in mammalian brains Hennig et al. [2021] in DNNs. It has been observed that the spiking activity of biological neurons in the same brain region tends to be correlated Hennig et al. [2021], Sadtler et al. [2014] and, the structure of this correlation tends to persist over long periods of time even if it limits performance and learning Golub et al. [2018]. In contrast, the activations of DNN neurons in the same layer are conditionally independent of each other given the outputs of the previous layer, and thus are not constrained in this way. This makes it possible for an adversary to induce arbitrary activation patterns Paiton et al. [2020], including those that lead to misclassification. Our experimental results demonstrate that adding this constraint does improve robustness of image and speech classification DNNs to adversarial attacks and non-adversarial perturbations.

Another cognitive mechanism that we study is recurrent connectivity in the biological brain. Most modern DNNs, particularly those that are commonly employed for computer vision applications, process the input in a feed-forward manner – each neuron in a layer receives inputs only from neurons in the previous layer(s). On the other hand, in the primate visual system, the neurons are connected in a highly recurrent manner – neurons may receive inputs from neurons either the same, any preceding or any succeeding visual area [Bullier et al., 2001, Briggs, 2020]. This recurrent processing has been linked to the ability of primates to perform accurate object recognition under distortions such as crowding and occlusions [Spoerer et al., 2017]. We extend this body of work by integrating recurrent circuits between neurons in the same layer (lateral recurrence), as well as between neurons from different layers (feed-back recurrence). We train these models on image classification tasks, with the additional objective of in-painting randomly placed occlusions. Experimental results show that doing so results in improved adversarial robustness.

1.5 Audio Robustness Benchmark

We have developed an audio robustness benchmark for the purpose of evaluating a DNN's robustness to adversarial and non-adversarial perturbations. While such benchmarks exist for vision tasks Hendrycks and Dietterich [2019], they do not for audio and speech tasks. Currently, different studies take different views of robustness and consequently evaluate it using different methods. For example, Hsu et al. [2021b], Likhomanenko et al. [2020] somewhat equate robustness with domain transfer, and evaluate robustness by computing the error rate of Automatic Speech Recognition (ASR) on a variety of datsets. This approach however is very coarse, and does not provide fine-grained information on what types of perturbations do the DNNs struggle against. On the other hand, Radford et al. [2023] also evaluates on room impulse responses, and environmental

and white noise. While better than the former approach, this method still does not encompass the variety of perturbations that humans are robust to and does not include adversarial attacks. To remedy this situation we develop a comprehensive audio robustness benchmark comprising over 100 challenging speech recognition scenarios that models are likely to encounter in the real-world.

1.6 Conclusion

In this thesis, we work towards DNNs that are robust to a variety of adversarial attacks by identifying principles, or priors, that can endow DNNs with robustness to adversarial attacks, without training them on adversarially perturbed data. In this connection, we have studied priors over the design elements and the structure of DNNs (structural priors) as well as priors derived from biological perception (biological priors) that seek to simulate biological mechanisms and constraints considered to be conducive to robustness. The contributions of our work are summarized as follows:

- We have developed a technique for removing the influence of spurious features from DNNs by pruning neurons that encode them, and demonstrated that it improves the adversarial robustness of DNNs, even surpassing adversarial training in cases where the test and training adversarial attacks differ. (Chapter 3)

- We have devised a framework for empirically estimating the probability that gradient descent, starting from random initialization, can converge to an adversarially robust decision boundary. We use this framework to identify the structural priors that increase this probability and consequently induce adversarial robustness in DNNs. (Chapter 4)

- We have created an image filter, *R-Blur* that simulates foveation via adaptive Gaussian blurring, and color desaturation. We demonstrate that models augmented with *R-Blur* retain most of the high classification accuracy of the base DNN while being more robust to both adversarial and non-adversarial image corruptions. (Chapter 5)

- We simulate the phenomenon of inflexible inter-neuron correlations observed in mammalian brains in DNNs, and show that doing so improves robustness to adversarial and non-adversarial perturbations. (Chapter 6)

- We introduce recurrent feedback connections in DNNs and demonstrate that adversarial robustness of the DNN is improved when these connections are optimized to reconstruct the input. (Chapter 7)

- We evaluate the impact of existing biologically plausible audio feature extraction methods, such as chochleagrams, as well our novel methods for simulating the phenomenon of frequency masking, and lateral suppression, on the adversarial robustness of speech recognition systems. (Chapter 8)

- Finally, we develop a comprehensive audio robustness benchmark comprising a variety challenging speech recognition scenarios that simulate real-world conditions that we would like the DNNs to be robust to. (Chapter 9)

The outcomes of our studies indicate that we *can* endow DNNs with a degree of generalized adversarial robustness by incorporating certain robustness priors related to the architecture and feature representations of the DNN, and without training them on a variety of adversarial attacks. This represents a step taken towards developing models that retain accuracy in the face of a variety of adversarial attacks and thus can be safely and reliably deployed in real-world settings. Our studies also reveal that deriving these priors from biology is a promising direction, and one that allows us to leverage the optimizations performed by evolution over millennia that have endowed humans and other primates with robust perception. The fact that integrating biological priors indeed endows DNNs with generalizable robustness indicates that doing so bridges some of the gaps between DNNs and biological perception, at least so far as robustness is concerned. In a word, these results provide evidence that there are more practical and scalable alternatives to the dominant approach of seeking robustness via training, and that DNNs with high accuracy and generalized adversarial robustness are, in fact, within reach. We envision that future research directions stemming from our work will further expand the body of structural and biological robustness-enhancing priors as well as discover other types of priors, particularly those related to the optimization algorithms used to learn DNN parameters.

Chapter 2

Background and Related Work

In this chapter, we scope our work and present the most closely related work. Work on DNN robustness falls in the following main categories: 1) Adversarial Attacks, 2) Defenses against adversarial attacks, and 3) Techniques for measuring robustness. Our work primarily falls under the second category, but we also provide robustness benchmarks for speech which aids in measuring robustness and thus falls under the third category. We discuss some of the related work in each of the aforementioned categories belowand position our thesis in this broad plethora of work.

2.1 Adversarial Attacks Against DNNs

Adversarial perturbations are perturbations that can change the response of DNNs when added to their inputs but are either imperceptible to humans or perceptually and semantically irrelevant enough to be ignored by them. Formally, adversarial perturbations for a given DNN, Φ, and a given input x, with ground truth label y_x, form the set

$$\mathcal{P}_{adv} = \{\delta | \delta \in \mathcal{I}_x \wedge \Phi(x+\delta) \neq \Phi(x)\} \tag{2.1}$$

where $\mathcal{I}_x$ is the set of perturbations of x that are imperceptible or semantically irrelevant. It is generally assumed that $\Phi(x) = y_x$ because otherwise the competence of the DNN is suspect and evidence of pathological behavior is less concerning.

Algorithms for generating adversarial perturbations are known as *adversarial attacks*. Usually, adversarial attacks solve the following optimization problem to generate an adversarial perturbation $\delta \in \mathcal{P}_{adv}$ [Szegedy et al., 2014, Goodfellow et al., 2014, Madry et al., 2018b]:

$$\delta_x = \arg\max_{\delta \in \mathcal{I}_x} \mathcal{L}(\Phi(x+\delta), y_x), \tag{2.2}$$

where $\mathcal{L}$ computes some metric of divergence. This type of attack is *untargeted*, where the purpose of the attacker is simply to cause a misclassification and is analogous to a denial-of-service attack. Alternatively, attacks can also be designed to cause *targeted* misclassification by solving the following optimization problem:

$$\delta_x = \arg\min_{\delta \in \mathcal{I}_x} \mathcal{L}(\Phi(x+\delta), y_t), \tag{2.3}$$

where y_t is the target label that the attacker wants x to be classified as. The various adversarial attacks proposed over the years generally differ in their choice of optimization technique and their characterization of $\mathcal{I}_x$.

Adversarial attacks have leveraged various optimization techniques over the years. The choice of optimization technique is, to an extent, determined by the level of knowledge about the target DNN the attacker is assumed to have (i.e. the *threat model*). Attacks that assume full knowledge of the target model's architecture and weights(*white-box* threat model) often use gradient-based optimization techniques, such as Limited Memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS) [Szegedy et al., 2014], Fast Gradient Sign Method (FGSM) [Goodfellow et al., 2014], and Projected Gradient Descent (PGD) [Madry et al., 2018b], to optimize δ (or parameters of a function that generates δ [Laidlaw et al., 2021]). If it is assumed attacker does not have any knowledge of the target DNN's architecture and only have query access to it (*black-box* threat model), adversarial attacks resort gradient-free optimization methods [Wang et al., 2022a] like random search [Andriushchenko et al., 2020, * et al., 2018], Mote-Carlo Tree Search [Wicker et al., 2018], finite-differences [Chen et al., 2017, Zhao et al., 2020] and evolutionary search Vo et al. [2022]. Black-box attacks, naturally, generally succeed less often than white-box attacks, and tend to involve querying the target DNN a very large number of times [Andriushchenko et al., 2020, Chen et al., 2020a], which can make them inefficient or even infeasible in certain cases.

Scenarios between white and black box threat models (*grey-box* threat model) also exist, when the attacker does not have access to the target model's parameters but may have some information about the target DNN's architecture, and training setup. In such situations the attacker may be able to avoid the pitfalls of black-box attacks by leveraging the transferability of adversarial attacks [Papernot et al., 2016]. To do this the attacker can use white-box attacks to generate adversarial perturbations that succeed on a *proxy* DNN and apply them to the inputs of the target DNN.

The other key element of adversarial attacks is how $\mathcal{I}_x$ is characterized. Several methods have been proposed over the years to characterize $\mathcal{I}_x$ such that it approximates the boundaries of human perception, which themselves are not precisely known. Some attacks limit the region to which the attack can perturb by specifying the shape, area, or objects to which the perturbation may be added Sharma et al. [2022]. Other attacks have used DNN-based approximations of human perception Qin et al. [2019], Laidlaw et al. [2021] or sophisticated distance metrics like Wasserstein distance [Wong et al., 2019b, Wu et al., 2020] to constrain δ. However, by far the most popular approach has been to use ℓ_p norm bounds to limit the size of the perturbations [Goodfellow et al., 2014, Madry et al., 2018b]. Each of these approximations yields different distributions of perturbations, therefore, models that are robust to one type of perturbation may not be robust to others Kang et al. [2019], Laidlaw et al. [2021].

2.1.1 Attack Algorithms

In the following sections we provide details about adversarial attacks that we have extensively used in this thesis.

Projected Gradient Descent (PGD) Attacks

Madry et al. [2018b] proposed an attack that used PGD as the optimization technique to solve equation 2.2 (or 2.3) and generate adversarial perturbations. PGD is an iterative algorithm, which

performs the two operations in each iteration: first, a gradient ascent step updates δ_x, and then a projection operator, $\Pi_{\mathcal{I}_x}(.)$, projects $\mathcal{I}_x$ onto $\mathcal{I}_x$. Formally, these steps can be written as follows:

$$\delta_x \leftarrow \delta_x + \eta \nabla_{\delta_x} \mathcal{L}(x + \delta_x, y_x) \quad \text{(gradient ascent)}, \tag{2.4}$$

$$\delta_x \leftarrow \Pi_{\mathcal{I}_x}(\delta_x) \quad \text{(projection)} \tag{2.5}$$

Several improvements to the original PGD attack have been proposed to improve its effectiveness. These include the addition of momentum [Dong et al., 2018], multiple restarts [Uesato et al., 2018], and adaptive step sizes [Croce and Hein, 2020c].

AutoAttack

AutoAttack Croce and Hein [2020c] is an ensemble of 4 adversarial attacks: untargeted and targeted Auto PGD (APGD) Croce and Hein [2020c], targeted Fast Adaptive Boundary (FAB) attack [Croce and Hein, 2020b], and untargeted Square attack [Andriushchenko et al., 2020]. APGD is a refinement of the PGD attack which adds momentum to the gradient updates and adaptively reduces (halves) the step size (η in equation 2.4) if there is no improvement in the attack objective. The FAB attack finds the smallest perturbation (under ℓ_p-norm constraints) that causes misclassification, and the Square attack is black-box attack that uses random search to perturb square regions of random widths at random locations until the input is misclassified. Ensembling attacks with diverse objectives and optimization techniques yields an attack that is stronger than its components, and thus gives a more accurate picture of the DNN's robustness. Furthermore, it is known that the presence of non-differentiable components in DNNs can render gradient-based attacks ineffective [Athalye et al., 2018a], but gradient free attacks may still be able to succeed. Therefore, the inclusion of the black-box Square attacks ensures that the robustness of such models is accurately measured.

2.2 Defenses Against Adversarial Attacks

A variety of methods have been proposed over the years to defend DNNs against adversarial attacks. In the following we discuss four categories of defenses against adversarial attacks that represent the majority of the prior work Akhtar et al. [2021], namely detection, input transformations, certified defenses, and adversarial training. To these we add two categories of defenses that are most relevant to our work, namely defenses that integrate structural and biological priors into DNNs to enhance their robustness.

Adversarial training, which is perhaps the most successful class of adversarial defenses, trains models on adversarially perturbed data generated by backpropagating gradients from the loss to the input during each training step. Madry et al. [2018b] formalized adversarial training as the following min-max optimization problem:

$$\Phi_{\text{robust}} = \arg\min_{\Phi} \mathbb{E}_{(x,y_x)} \left[\arg\max_{\delta_x \in \mathcal{I}_x} \mathcal{L}(\Phi(x + \delta_x), y_x) \right], \tag{2.6}$$

where the inner optimization is performed by the PGD attack, and the outer optimization is performed by stochastic gradient descent. Practically this amounts to running a few iterations of PGD

update on each training batch δ_x such that the loss is increased, before using SGD on the final loss value to update the parameters. Several refinements of Madry et al. [2018b]'s approach have been proposed to increase its effectiveness and efficiency Zhang et al. [2019], Rebuffi et al. [2021], Bai et al. [2021], Wong et al. [2019a]. The key shortcoming of this approach is that it fails to generalize to unseen attacks, i.e. attacks that use different optimization methods or approximations of $\mathcal{I}_x$ [Akhtar et al., 2021, Song et al., 2019, Geirhos et al., 2019, Maini et al., 2020]. We would like to point out that some studies have proposed to improve the generalizability of adversarially trained models by applying multiple types of adversarial attacks during training Maini et al. [2020]. We believe that this approach does not solve the fundamental problem because it still yields models that are robust to the attacks used during training and does not generalize to unseen attacks.

Certified defenses Cohen et al. [2019a], Fischer et al. [2020], Kumar and Goldstein [2021], Li et al. [2019] are a class of defenses that also provide provable guarantees of the form: with probability $1 - \alpha$, the model's output will not change by perturbation of size at most ϵ. A common technique used by certified defenses is randomized smoothing Cohen et al. [2019a], Salman et al. [2020], Cao and Gong [2017] which perturbs the input by multiple randomized perturbations (with a known distribution) and aggregates the models predictions over them. Since the distribution of the perturbations is precisely known, theoretical guarantees of the aforementioned form can be obtained. Other techniques used by certified defenses include regularization [Croce and Hein, 2020a], and convex relaxation Wong and Kolter [2018]. The theoretical guarantees make certified defenses attractive in security-critical applications where it is useful to quantify the risk. However, computing these guarantees requires the exact nature of the perturbations to be known and the proofs have to be redone for each type of perturbation. This makes this type of defense infeasible for defending against diverse and unseen perturbations. Nevertheless, we use the certification procedures that this body of work proposes to verify the robustness of the defenses proposed in this thesis on known perturbation types.

Input transformation based defenses seek to thwart adversarial attacks by applying transforms to the input during inference. Earlier works [Guo et al., 2017] used simple transforms like JPEG compression, bit-depth reduction, total variance minimization, and image quilting. However, such defenses were shown to be weak because they relied on the non-differentiability of transforms to render gradient-based attacks ineffective, and thus could be bypassed by gradient-free attacks, as well as more advanced gradient based attacks Athalye et al. [2018a]. Recent works apply a multitude of random transformations to the image [Raff et al., 2019]. While the transforms themselves are rather simple, their number, order and parameters are randomized. This procedure does not make the DNN itself more robust, and it has been shown that increasing the number of optimization steps of PGD-based attacks and/or using techniques like Expectation-over-Transformations (EoT) Athalye et al. [2018b] can marginalize out the randomness and allow adversarial attacks to succeed. Such marginalization techniques, however, impose significant computational costs Sitawarin et al. [2022] and thus the defense can still be considered feasible against resource constrained adversaries.

Detection base defenses, like Metzen et al. [2017], add additional modules to the DNN to detect the presence of adversarial perturbations. Inputs flagged as adversarial may be rejected or processed by a different pathway than benign inputs. This class of defenses has fallen out of favor in the community Akhtar et al. [2021] because they do not solve the core problem that DNNs are sensitive to imperceptible and irrelevant perturbations.

To summarize, the aforementioned categories of adversarial defenses have significantly im-

proved the robustness of DNNs to existing adversarial attacks, allowing DNNs to retain significantly more prediction accuracy than undefended models Madry et al. [2018b], Cohen et al. [2019a], however, defenses in these categories do not generalize across existing perturbations types, let alone unseen ones. The development of defenses that robustly generalize is a growing field of research. Given that the work presented in this thesis also fall within this field, in the following we review prior work on robust generalization, with a focus on works that study structural and biological robustness priors.

Structural modifications that improve robustness have been explored in prior work, however, many of these works seek to improve the effectiveness of adversarial training by making structural modifications [Akhtar et al., 2021, Huang et al., 2023]. Guo et al. [2020] uses neural architecture search, guided by performance on adversarial attacks to design DNNs. Guo et al. [2018] propose robustness guided DNN pruning. Huang et al. [2021] conduct an empirical study on the impact of DNN depth and width on the effectiveness of adversarial training. While these approaches are related our work on structural robustness priors in so far as they also seek robust DNN architectures, and the removal of DNN components that enable adversarial attacks, they differ fundamental from our approach because we seek structural priors that enhance the robustness of DNNs *trained only on natural data*. Furthermore, since adversarial perturbations are part of the loop in these methods, they share the pittfalls of adversarial training vis-a-vis generalization and efficiency.

Prior work has also explored methods of discovering structural robustness priors are methods that do not involve adversarial perturbations during training. Fu et al. [2021] shows that subnetworks within randomly initialized networks exist that exhibit adversarial robustness on par with adversarially trained DNNs. Wang et al. [2018] frame the problem of adversarial robustness as arising from an over-specified input space whose true rank is much lower than its dimensionality, which gives rise to spurious correlations. They posit that compressing and quantizing the input will reduce the over-specification and improve robustness. Other methods propose to add regularizations, such as bit-plane consistency [Addepalli et al., 2020], Jacobian-based GAN-like regularization [Chan et al., 2019], and gradient phase and magnitude regularization [Dabouei et al., 2020]. Some works Feather et al. [2023], Wang et al. [2020] have tried to modify the receptive fields of the convolution kernels to improve robustness.

Perhaps more relevant to our work are studies that systematically evaluate the impact on robustness of various attributes of DNNs, such as width, depth, normalization and initialization. Several papers Loo et al. [2022], Patane et al. [2022], Bubeck and Sellke [2021], Zhu et al. [2022] have sought to determine the impact of DNN width on adversarial robustness using neural tangent kernels, which are analogous to infinite width DNNs. Benz et al. [2021] discovered that batch normalization leads to reduced adversarial robustness due to the mismatch of the mean and variance statistics of natural and adversarially perturbed data. Zhu et al. [2022] conduct a systematic analysis of the relationship between perturbation stability of DNNs and their width, depth, and initializations, and provide theoretical results showing that increasing width also increases robustness, while increasing depth may improve robustness or cause it to deteriorate depending on the parameter initialization. The study of Zhu et al. [2022] complements our work on discovering structural priors ([REF SEC]), however, while the theoretical analysis in this study is limited to certain types of elements of DNNs, our empirical technique is not limited in this way and can be used to discover a wider range of structural priors for robustness.

Several **biological defenses** have been proposed over the years. These defenses involve integrating computational analogs of biological processes that are absent from common DNNs. The

resulting models are made more robust to adversarially perturbed data, and have been shown to better approximate the responses of biological neurons Dapello et al. [2020]. Most relevant to our work are approaches related to foveation, lateral and top-down recurrence, and early stages of audition. We present prior work related to each of these below.

Foveation refers to the phenomenon of biological vision that causes the fidelity of the image to be maximal close to the point of fixation, and lower in regions further away from it. In an early work, Luo et al. [2015] investigates the impact of foveation on adversarial robustness. They implement foveation by cropping the salient region of the image at inference time and show reduction in attack success rates. This work has several shortcomings. Firstly, the biological plausibility of this method is questionable because it does not simulate the degradation of visual acuity in the periphery of the visual field, rather it discards the periphery entirely. Secondly, it crops the image after applying the adversarial attack, which means that the attack does not take into account the cropping, which is akin to obfuscating the gradients, and hence any reported improvements in robustness are suspect. A later work Vuyyuru et al. [2020] (Retina Warp) avoids the aforementioned pitfalls and simulates foveation via non-uniform sampling (regions further away from the fixation points are sampled less densely). Since this method is fully differentiable and highly biologically plausible, we compare against it in this thesis. Some recent works Jonnalagadda et al. [2022], Gant et al. [2021] apply foveation in the latent feature space (the intermediate feature maps generated by a CNN). These works implement foveation by changing the receptive field sizes of the convolutional kernels based on the distance to the fixation. Since they operate on the latent feature space, rather than image pixels, their methods not directly comparable to our work in Chapter 5.

While **Biologically derived acoustic features** [Feather et al., 2019, Kim and Stern, 2016, Slaney, 1988, Lyon, 1984, Ghitza, 1986, Seneff, 1988, Hermansky et al., 1991] have indeed been extensively studied in prior works, they are not widely used in modern speech processing systems, which generally operate on either the raw waveform [Baevski et al., 2020] or the log mel spectrogram [Radford et al., 2023]. While the use of the Mel filterbank does impart a degree of biological plausibility to the log Mel spectrogram, it does not represent several processes that occur in the cochlea and the auditory nerve, including filtering, more realistic non-linearity, lateral suppression and cross/auto-correlation [Stern and Morgan, 2012]. Prior research has shown that while more bio-plausible feature extraction methods[Stern and Morgan, 2012, Lenk et al., 2023] do not usually improve speech recognition performance on clean speech, they do improve performance on degraded and noisy speech, however, their impact against adversarial perturbations is yet to be fully evaluated.

Recurrence is known to be prevalent in the human and, in general, animal brain [Bullier et al., 2001, Briggs, 2020], and has been linked to the robustness of both audio Stern and Morgan [2012] and visual Spoerer et al. [2017], Wyatte et al. [2014] perception. In contrast, most DNNs used for perceptual tasks, particularly image-based ones, are predominantly feed-forward. While Recurrent Neural Networks (RNN) are widely used in speech recognition, they add recurrent circuits between groups of neurons in the same layer (analogous to brain regions). Meanwhile, recurrent circuits in the brain also connect neurons from different regions, resulting in feed-back circuits carrying information from areas of higher-level processing to those of lower-level processing. Markov et al. [2014] posits that feedforward DNNs can model human visual perception immediately after ($<$ 200 ms) the onset of the stimulus, because when the stimulus is presented for an extremely short duration, humans tend to make similar errors as DNNs. In the case of auditory perception, feed-back circuits orginating in higher-processing areas go all the way back to the cochlea, which

leads to amplification of lower amplitude signals, and compression of the dynamic range Stern and Morgan [2012].

Given the relationship between robust perception and recurrence, several approaches have been proposed to integrate recurrence into DNNs. One of the most well known of these is known as the predictive coding hypothesis Rao and Ballard [1999], which posits that recurrent connections are involved in a form of Bayesian optimization in which the feed-forward activations are optimized to be maximally predictive of the observed stimulus. Paiton et al. [2020] analyzed the dynamics of predictive coding to show that it has the effect of pushing the hidden representations of the DNN to prototypical representations, thus limiting the ability of an adversarial attack to induce arbitrary representations. Choksi et al. [2021] and Huang et al. [2020] propose methods for integrating predictive coding in DNNs and demonstrate improved robustness to adversarial attacks and other corruptions. Our work on constraining the inter-neuron correlations (Chapter 6) can be seen as a special case of predictive coding in which the activations of the neurons are not only optimized to be predictive od the input, but also to respect a fixed inter-neuron correlation matrix that is learned during training.

There is a body of literature on applying recurrent DNN architectures that departs from the predictive coding hypothesis and seek to train recurrent models without specific constraints. Schwarzschild et al. [2021] and Bansal et al. [2022] developed recurrent models maze solving DNNs that have the ability to dynamically adjust their analysis depending on the complexity of the task, and even to correct for corruptions. Kubilius et al. [2018] demonstrate that adding recurrence to image recognition DNNs improves their correspondence to biological neurons. These prior works mostly added lateral recurrence, i.e. recurrence between neurons in the same layer (or block). Our work on integrating recurrent connections to DNNs extends this body of work by introducing feed-back connections.

2.3 Robustness Evaluation

Prior work has developed guidelines and benchmarks for accurately evaluating the robustness of DNNs and reliably tracking progress by enabling fair comparisons between methods. In the domain of vision, the following works have proposed guidelines and benchmarks. Carlini et al. [2019] provide guidelines on how to reliably measure the adversarial robustness of DNNs. Croce et al. [2020] propose an adversarial robustness benchmark and leaderboard based on AutoAttack Croce and Hein [2020c]. Hendrycks and Dietterich [2019], Hendrycks et al. [2021a] and Hendrycks and Dietterich [2019] propose benchmarks and metrics for measuring the robustness of image recognition models to non-adversarial perturbations. In the domain of audio and speech processing however there is considerably less prior work on robustness benchmarks. To measure adversarial robustness of speech recognition models Olivier and Raj [2022b] developed a library containing implementations of a variety of adversarial attacks, which they applied to a number of recent speech recognition models and presented the results. While this work is indeed a very good starting point, it is not, nor does it claim to be, a benchmark insofar as it does not propose a particular evaluation methodology. The measurement of non-adversarial robustness in prior work has been done in a non-standardized way with each paper defining and measuring robustness differently. A common definition seems to equate robustness with domain transfer Likhomanenko et al. [2020], Radford et al. [2023], Hsu et al. [2021b]. Practically this means evaluating on a variety of datasets, under

the assumption that these datasets sufficiently represent the real world diversity. This approach has two major shortcomings: (1) the assumption that commonly available datasets would accurately reflect real world diversity may not be true, and (2) this evaluation methodology does not provide fine-grained results about the strengths and weaknesses of DNNs; for example, Likhomanenko et al. [2020] shows word error rates for 3 settings: clean, noisy, and extreme, for each dataset. It is not readily discernible exactly what types of perturbations the model has issues with. Our work on the audio robustness benchmark aims to fill this gap in prior work.

Part I

Structural Robustness Priors

Chapter 3

Towards Adversarial Robustness via Compact Feature Representations

3.1 Problem and Motivation

Adversarial attacks exist, in large part, because DNNs are overly sensitive to spurious features Ilyas et al. [2019], such as high-frequency components Wang et al. [2020] and texture Geirhos et al. [2018a]. These features are somewhat correlated with the true labels, either coincidentally or due to sampling biases in the training data, however, they are often semantically unrelated to the ground truth, and meaningless for humans Wang et al. [2020]. Therefore an adversary can craft adversarial perturbations that modify these spurious features and induce misclassifications in DNNs, while remaining invisible or irrelevant to humans Geirhos et al. [2018a]. Hence, reducing the model's reliance on superfluous features can make it more robust to adversarial perturbations.

In this thesis, we propose a method to remove the influence of superfluous features from the model. The cornerstone of our approach is the observation that each neuron in the hidden layers of a neural network is a feature detector for the subsequent subnetwork. Based on this observation, we can drawn an equivalence between neurons and features, which implies that to remove the influence of a superfluous feature we must remove the neuron that detects it. To identify such neurons we decompose the features learned by a neural network into two components: the redundant information that is already encoded by other features, and the novel information encode by this feature. Based on this decomposition we select neurons that provide the maximum amount of novel information about the target output and discard neurons that encode redundant or irrelevant information. We modify a neuron pruning technique called LRE-AMC Shah et al. [2021] to use our neuron selection criterion and remove superfluous neurons from several well known image recognition models. We discuss the technical details of our approach below.

3.2 Technical Overview

3.2.1 Redundant Features Allow Adversarial Attacks

Central to our study is the observation from Wang et al. [2018] that an overspecified input is a necessary condition for the existence of adversarial examples for Machine Learning (ML) models.

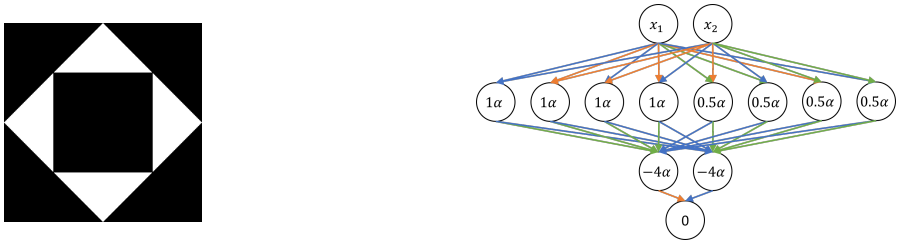


Figure 3.1: a) The example decision boundary. b) A minimal network that models it. The blue, orange and green connections represent weights of +1,-1, and 0. The numbers in the circles represent the biases. (Image credits Shah and Raj [2020])

ML models operating on over-specified inputs need to be able to model the target function for a larger number of input patterns compared to ML models operating on minimally specified inputs. For example, a model can learn the function $y = x1XORx2$ with just four training patterns. However, if the input to the model is overspecified by adding an additional variable, $x3$, it would need to see eight input patterns to learn that the value of x_3 has no impact on the output y. If some input patterns were not available to the model during training (as is usually the case), the model would need to choose, from an exponential number (in the number of missing patterns) of possible assignments, one assignment of y to associate with the missing patterns.

If the model learns an incorrect assignment of y it would be possible for an adversary to change the output of the model by modifying x_3.

We can consider x_1, x_2 and x_3 to be *features*, and thus, to make models more robust we would want them to use the smallest set of features that can be used to correctly compute the target function, i.e. the Minimal Sufficient Statistic (MSS).

3.2.2 Neurons Are Features in a Deep Neural Network

In a DNN neurons in each layer compute features from the layer's input so the number of features computed in a layer is equal to the number of neurons in that layer (assuming distinct and non-zero activations). These features then become inputs for the downstream network. To illustrate this, consider the simple decision boundaries presented in Figure 3.1(a) that are modeled perfectly by the handcrafted network with threshold activations in Figure 3.1(b). The eight neurons in the first layer are feature detectors for the eight boundaries. Each neuron indicates on which side of the respective boundary the input point lies. The two neurons in the second layer determine if the input point is located within each of the two squares *using the features derived from the input in the first layer*. Likewise, the output neuron in the final layer is a linear classifier that operates on the features computed by the second layer.

Given the equivalence between features and neurons, if a layer has too few neurons, the input to the downstream subnetwork would be under-specified causing the model to perform poorly. If the layer has too many neurons the input to the downstream network is overspecified making it vulnerable to adversarial attacks. In the subsequent section we will present a method for reducing the number of superfluous neurons (features) in the model, and thereby reducing the adversary's attack surface and making the model more robust.

3.2.3 Identifying Non-Spurious Features

Suppose that the model (or a layer in the model) computes the feature set $\mathcal{F} = \{f_1, ..., f_n\}$. Let $\mathcal{F}_{-i} = \mathcal{F} \setminus \{f_i\}$. Each feature $f_i \in \mathcal{F}$ can be decomposed into two components as

$$f_i = \eta_i + \delta_i \tag{3.1}$$

where the η_i is predictable from the *other* features in $\mathcal{F}$ i.e. $\eta_i = \phi_i(\mathcal{F}_{-i})$, while δ_i is the residual information carried by the feature. In our current implementation we take ϕ_i to be a linear function. The details of this estimation are presented in 3.3.1.

Based on this decomposition we can define the "usefulness" of a feature (for a given task) as *the amount of residual information it carries about the target output*. In information theoretic terms, the usefulness of f_i can be quantified as, $I(\delta_i; Y)$, the mutual information between the residual, δ_i, and the target output Y. Computing mutual information for random variables with unknown distributions is challenging so we approximate it. While there may be others, in this paper we explore two methods to perform this approximation:

1. Estimating the effect of removing δ_i from f_i on the loss function using a first-order taylor approximation around f_i as

   $$\Delta_{L_i}(\delta_i) = -L_i'(f_i)\delta_i \tag{3.2}$$

   where $L_i(f_i)$ represents the loss when feature i is f_i.

   This provides an efficient method for computing Δ_{L_i} because apart from δ_i, all the other required information is already computed during forward and back-propagation steps.

2. Estimating a lower bound on mutual information using MINE Belghazi et al. [2018]. MINE is a neural model that computes the neural information measure, $I_\Theta(X, Z) \leq I(X, Z)$. We estimate the non-spurious information encoded by a feature as $I_\Theta(\delta_i, Y)$.

After having identified the neurons that compute spurous feature, we proceed to remove them using Annealed Model Contraction with Lossless Redundancy Elimination (LRE-AMC) Shah et al. [2021].

We have chosen LRE-AMC because its neuron selection criteria is closely related to the one we proposed above except that it prunes away neurons with the smallest values of δ_i, whereas we use the mutual information between δ_i and the target output Y. This similarity greatly simplifies the implementation of our proposed neuron selection criteria from 3.2.3, especially of equation 3.2. In the next section we briefly describe LRE-AMC and provide further details about our modifications.

3.3 Removing Spurious Features Using Annealed Model Contraction

3.3.1 Annealed Model Contraction with Lossless Redundancy Elimination (LRE-AMC)

LRE-AMC Shah et al. [2021], Shah and Raj [2020], iteratively removes neurons from each layer and fine-tunes the compressed model using a knowledge distillation objective Hinton et al. [2015].

During pruning, LRE-AMC prioritizes the removal of neurons whose post-activation outputs are (almost) linearly predictable from the outputs of the other neurons in the same layer.

To identify linearly predictable neurons in layer l, LRE finds a transformation matrix, $A^{(l)}$, that approximates an identity map over $\mathbf{z}^{(l)}$, the non-linear activations of neurons in layer l. To exclude the trivial solution $A^{(l)} = I$, the diagonal components of A are contstrained to be zero. Formally $A^{(l)}$ is defined as

$$A^{(l)} = \min_{A} \mathbb{E}_{\mathbf{z}^{(l)}=f_{1:l}(x)|x\sim D} \left\| \mathbf{z}^{(l)^T} A^{(l)} - \mathbf{z}^{(l)} \right\|_2^2 \; s.t \; diag(A) = 0 \tag{3.3}$$

The columns of $A^{(l)}$ contain the coefficients that best predict the corresponding component of $\mathbf{z}^{(l)}$ as a linear combination of the all the other components. We can use A to estimate $\eta_i^{(l)}$ as $\eta_i^{(l)} \approx \mathbf{z}^{(l)^T} A_{.i}^{(l)}$ and $\delta_i^{(l)}$ as $\delta_i^{(l)} \approx \mathbf{z}_i^{(l)} - \eta_i$. Neurons whose post-activation output is linearly predictable, i.e. $|\delta_i|$ is small, can be safely removed since they are not contributing useful information to the model.

After each round of pruning the model is fine-tuned using the following objective

$$\mathcal{L} = (1-\alpha) H\left(\sigma\left(\frac{\mathbf{z}}{\tau}\right), \sigma\left(\frac{\mathbf{v}}{\tau}\right)\right) + \alpha H\left(\mathbf{y_{true}}, \sigma\left(\mathbf{v}\right)\right) \tag{3.4}$$

where H represents the cross-entropy, $\mathbf{z}$ and $\mathbf{v}$ are the logits from the unpruned and pruned models, respectively, σ is the softmax function and, τ and α, respectively, control the temperature of the distribution and the relative contributions of the two loss terms.

3.3.2 Modifying LRE-AMC for Removing Spurious Features

We incorporate the measures of spuriousness from 3.2.3 into LRE-AMC by modifying the scoring function it uses to select neurons for pruning. To compute the gradient based measure of spuriousness we scale δ_i by the empirical estimate of $L_i'(f_i)$ such that the score for the neurons in layer l is computed as

$$\boldsymbol{\delta}^{(l)} \odot \mathbb{E}_{\mathbf{z}^{(l)}=f_{1:l}(x)|x\sim D} \nabla_{\mathbf{z}^{(l)}} \mathcal{L}(\boldsymbol{\eta}^{(l)}) \tag{3.5}$$

where $\boldsymbol{\eta}^{(l)}$ and $\boldsymbol{\delta}^{(l)}$ are vectors containing the η_is and δ_is corresponding to $\mathbf{z}_i^{(l)}$s in $\mathbf{z}^{(l)}$, and $\odot$ is the elementwise product of vectors. Here we compute $\mathcal{L}$ with $\alpha = 1$ in equation 3.4. On the other hand, the mutual information based measure of spuriousness, $I_\Theta(\delta_i, Y)$, is directly computed using MINE. Apart from the scoring function, we do not modify any other aspect of LRE-AMC.

3.4 Key Results

In our experiments, we assume the worst-case whitebox threat model in which the adversary has complete access to the targeted model including the gradients for all the parameters. The adversary searches for adversarial examples in ℓ_∞ and ℓ_2 balls of various radii, denoted by ε, around the input using Projected Gradient Descent (PGD) Madry et al. [2017] on the cross-entropy loss. We used the implementation of PGD from Advertoch Ding et al. [2019] in our experiments.

We used CIFAR10Krizhevsky et al. and MNIST LeCun et al. [2010] datasets to train and evaluate several well-know deep learning models. Specifically, we trained VGG-16Simonyan and

Method	$-\Delta_P$	Acc_{cln}	$\mathbb{E}[\text{Acc}_{rob}]$	Acc_{rob} w/ $\|\varepsilon\|_\infty$			Acc_{rob} w/ $\|\varepsilon\|_2$		
				4	8	16	0.5	1.0	2.0
VGG16-CIFAR10									
None	0.0	90.3	1.3	1.4	0.0	0.0	4.0	1.8	0.6
None-AT	0.0	74.9	31.6	57.1	37.1	8.6	53.2	27.6	3.5
None-GS	0.0	82.9	20.6	43.5	13.8	0.8	47.6	16.6	1.0
TD-LG	87.7	85.6	17.7	20.0	17.4	13.3	20.6	19.3	15.8
TD-V	84.6	87.7	9.7	11.2	9.3	5.7	11.5	9.9	6.8
RR-MI	98.3	85.7	9.5	11.8	9.2	7.0	12.4	9.5	7.1
GL-MI	72.6	88.6	9.3	10.6	8.8	5.6	12.5	11.0	7.3
AlexNet-CIFAR10									
None	0.0	77.5	2.8	8.9	0.3	0.08	7.23	0.2	0.06
TD-V	97.7	74.6	18.6	23.7	17.0	14.3	25.0	17.3	14.5
RR-V	97.3	73.8	18.5	19.7	19.3	17.8	19.3	18.3	15.7
TD-LG	98.3	72.3	11.1	14.6	10.1	9.5	13.2	9.8	9.2
TD-MI	98.3	72.2	5.9	10.2	4.2	3.2	9.5	4.3	3.7
	$-\Delta_P$	Acc_{cln}	$\mathbb{E}[\text{Acc}_{rob}]$	Acc_{rob} w $\|\varepsilon\|_\infty$			Acc_{rob} w $\|\varepsilon\|_2$		
				0.2	0.3	0.4	2.0	3.0	4.0
LeNet-MNIST									
None	0.0	99.1	4.0	1.2	0.0	0.0	18.0	3.97	1.0
TD-MI	86.8	95.7	12.4	13.1	12.1	10.2	14.9	12.7	11.5
TD-LG	93.9	96.2	3.9	6.3	4.5	2.8	6.3	2.8	0.8
RR-MI	96.5	96.0	3.9	3.1	1.1	0.3	10.0	5.7	3.1

Table 3.1: The table presents the reduction in the number of parameters ($-\Delta_P$), the accuracy of the model on clean data (Acc_{cln}), the average accuracy of our CIFAR10 and MNIST models on adversarially perturbed data (Acc_{rob} w/ $\|\varepsilon\|_\infty$), and the accuracy of the model adversarially perturbed data for various perturbation sizes (Acc_{rob} w/ $\|\varepsilon\|$). The method follows the format [compression scheme]-[criteria].

Zisserman [2014] and AlexNetKrizhevsky et al. [2012] on CIFAR10 and LeNet LeCun et al. [1998] on MNIST. We trained the models from random initialization to convergence on the clean dataset using stochastic gradient descent, with a learning rate of 0.1 and ℓ_2 regularization weight of 0.005. During training 20% of the training data is used for validation, and if the validation accuracy does not improve for more than 5 epochs the learning rate is halved. The adversary performs 100 steps of PGD over ℓ_p balls. On CIFAR10 the adversary explored ℓ_∞ balls of $\varepsilon \in \left\{\frac{4}{255}, \frac{8}{255}, \frac{16}{255}\right\}$ and ℓ_2 balls of $\varepsilon \in \{0.5, 1.0, 2.0\}$ with steps size $\frac{\varepsilon}{4}$. In experiments on MNIST the adversary explored ℓ_∞ balls of $\varepsilon \in \{0.2, 0.3, 0.4\}$ and ℓ_2 balls of $\varepsilon \in \{2.0, 3.0, 4.0\}$ with steps size $\frac{\varepsilon}{40}$.

We configured LRE-AMC hyperparameters to be $\tau = 4$ and $\lambda = 0.75$. During the fine-tuning steps the Adam optimizer with learning rate 0.0001 and ℓ_2 regularization weight of 0.0001 are used. We experiment with three schemes of applying LRE-AMC to the model, namely Top-Down (TD), Round Robin (RR) and Global (GL). In TD we move up the model, starting from the penuiltimate layer, shrinking each layer maximally using LRE-AMC as long as the accuracy remains above a threshold, t. In RR, we repeatedly loop through the layers, starting from the top. In every iteration we apply LRE-AMC *once* to each layer and we repeat until no more neurons can be removed without degrading the model's accuracy beyond a threshold, t. In GL we score neurons from all

the layers in the network using the equations from 3.3.1 and use LRE-AMC to remove a fraction of the highest scoring neurons. In all cases, we configure LRE-AMC to remove up to 25% of the neurons from a layer (for TD and RR) or the whole network (for GL) in each pruning step. We conducted experiments with $t \in \{0.05, 0.03, 0.01, 0.0\}$. We run experiments with three scoring methods, specifically, the method from Shah et al. [2021] (V), the score based on the derivative of the loss described in equation 3.5 (LG) and the mutual information metric described in 3.2.3 (MI). We present our most salient results, in terms of the average accuracy on adversarial data, in Table. 3.1.

Our approach improves robustness to white-box attacks. We see that removing spurious neurons from the network using our approach greatly improves the model's robustness to ℓ_∞ and ℓ_2 bounded attacks, especially in the case of VGG16-CIFAR10 where using the gradient based selection criteria improved the average accuracy of the model, on perturbed data by more than 16% (absolute) compared to the standard model and by 8% compared to the model compressed with LRE-AMC. We observe similar improvements on LeNet-MNIST and AlexNet-CIFAR10. On AlexNet-CIFAR10 vanilla LRE-AMC outperforms our selection criteria. In fact, AlexNet with TD-V provides the best robust accuracy in all our experiments.

Comparison with Other Defenses: We applied two highly successful defences, namely adversarial training (AT) with PGD Madry et al. [2017] and gaussian smoothing (GS) Cohen et al. [2019b] to a baseline VGG16-CIFAR10. The results are presented in Table 3.1 as None-AT and None-GS. We performed AT using 7 PGD steps of size $\frac{2}{255}$ with $\|\varepsilon\|_\infty \leq \frac{8}{255}$. For GS we sampled noise from $\mathcal{N}(0, 0.25I)$. We note that for small values of $\|\varepsilon\|$ AT and GS defences make VGG16 significantly more robust than models pruned with our technique, however for larger values the accuracy of these models precipitously decreases. At $\|\varepsilon\|_\infty = 16$ and $\|\varepsilon\|_2 = 2.0$ our TD-LG model has an accuracy that is 4.7% and 12.3% greater than the AT model. In addition to being more robust at higher noise levels, the TD-LG model also achieves 9.7% greater accuracy on clean data than the AT model.

It appears that the defenses like AT and GS do not generalize very well to perturbations outside the ℓ_p ball they are trained with. We posit that a reason for this is that they do not explicitly evaluate the features being learned, and thus, it is possible that they retain some spurious features that adversaries with larger perturbation budgets can easily exploit. Furthermore, it is very important to note that, unlike the other methods, our models were trained on *clean images only*. The fact that our models were able to resist strong high-perturbation attacks without ever seeing out-of-distribution data suggests that the latter is not as strong of a requirement that existing studies have made it out to be. In fact, our results seem to suggest that techniques that rely minimally on perturbed training data can generalize better to different attack methods and configurations.

Chapter 4

Uncovering the Robustness Potential of Neural Architectures by Measuring the Probability of High Adversarial Accuracy

4.1 Problem and Motivation

Investigation into the nature of adversarial perturbations have suggested that ML models learn some features that are semantically irrelevant for humans but are sufficiently correlated with target output that by learning them the model can effectively minimize the training loss . These type of features have been called "non-robust features" because if the data was modified in such a way that *only* the non-robust features were perturbed, then to a human observer it would appear semantically identical to the original data, but it would appear different to a ML model that relies on these non-robust features. Conversely, features that are semantically relevant for humans can be called "robust features" because if they are perturbed the data will appear semantically different from the original data to a human. It follows that the non-robust features can exploited to generate adversarial perturbations and thus a model is robust to adversarial perturbations to the extent that it makes predictions based on robust features.

Broadly speaking there are two methods of training a ML model to learn a particular feature (or a particular class of features). The first method is to sample the training dataset such that without learning the desired feature the model will not be able to make accurate predictions. This approach will be successful to the extent that the training data represents the distribution of the desired feature and the model has sufficient capacity to learn the feature. Adversarial training [Madry et al., 2018a] uses this method by training the model on adversarially perturbed data instead of the original data. The second method is to introduce an inductive bias, which is usually based on domain knowledge, into the model itself or in its training algorithm that encourages it to learn the desired feature. For example, based on the knowledge that high-frequency components of images are not very relevant to visual perception (if they are even perceptible) Wang et al. [2019] showed that by smoothing the convolutional filter weights the robustness of a convolutional neural network may be improved. These two methods are not mutually exclusive and are often employed together, with the expectation that adding the inductive bias might reduce the amount of training data and the size of the model required.

In this thesis, we show empirically that common components and hyperparameter settings of Deep Neural Networks (DNNs) behave as inductive biases vis a vis the adversarial robustness of the DNN. We observe that if a DNN having sufficient capacity to model the robust boundary is trained to achieve high accuracy on clean data, it will, with a certain probability, also be robust – we refer to this probability as the *Natural Robustness Potential*(NRP) of the model. To estimate the NRP for a given DNN architecture and hyperparameter setting we train several DNNs, each having the same architecture and hyperparameters but different (random) parameter initialization and compute the NRP as the ratio of the number of models that achieved high accuracy on clean *and* perturbed data, and the number of models that achieved high accuracy on clean data, but not necessarily on the perturbed data.

We observe that manipulating common DNN design choices can influence the NRP of the resulting model. For instance, increasing the number of neurons or convolutional filters in each layer or adding shortcut connection increases the NRP of the model, while increasing the number of layers or adding batch normalization reduces the NRP of the model. Furthermore, we find that these observations generalize across datasets and models of different complexities – the trends observed in Multi-Layered Perceptron (MLP) models trained on a 2D toy task of computing a real valued XOR, can also be observed in MLPs and CNNs trained on the much more complex MNIST dataset. Changes in NRP induced by modeling choices are not chance occurrences or artifacts of the simple training data and small models, but rather indicate that certain DNN design choices in fact improve the likelihood of finding robust models independent of the training task and model complexity.

4.2 Technical Overview

To illustrate the phenomenon of accidental robustness we consider deep neural networks trained to approximate a real-valued variant of the XOR function. The Boolean XOR function (BXOR) is a binary operator (usually denoted $x_1 \oplus x_2$) that takes Boolean inputs and outputs True if and only if one of the inputs is True and the other is False. The function can also be extended to an arbitrary number of inputs by using the base case $\text{BXOR}(x_1) = x_1$ and the following recursive rule for $n \geq 2$:

$$\text{BXOR}(x_1, .., x_n) = \text{BXOR}(x_1, .., x_i) \oplus \text{BXOR}(x_{i+1}, .., x_n)$$

Unlike the case of, for example, functions defined over natural images, the BXOR domain spans the entire input space. Therefore while on image classification there is for each input a region of input space in which the pixel values may change semantics of the input remain unchanged; in the case of the Boolean XOR any perturbation changes the input in a semantically meaningful way. Due to this property, the Boolean XOR function does not permit adversarial perturbations as defined in this work. To get around this limitation, in this paper we use a real-valued variant of the XOR function (RXOR), which defined as $\text{RXOR}(x_1) = \text{sign}(x_1)$ and for $n \geq 2$:

$$\text{RXOR}(x_1, .., x_n) = \text{RXOR}(x_1, .., x_i) \neq \text{RXOR}(x_{i+1}, .., x_n)$$

where $x_i \in [-1, 1]$. It is easy to see that this formulation of the XOR function permits perturbations that do not change the input in semantically meaningful ways. For instance, if $x_{j \neq i}$ are fixed, changing x_i from 1 to $\epsilon > 0$ does not change the value of $\text{RXOR}(x_1, .., x_i, ..., x_n)$.

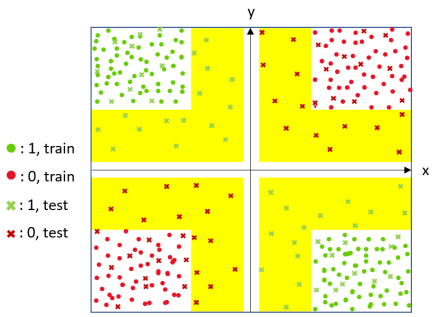


Figure 4.1: 2D RXOR dataset illustration. The yellow section is the adversarial region, i.e. in our framework set difference between the test data support and the training data support.

The domain of RXOR contains infinitely many elements, therefore it is impossible to train a model on the entire input domain. Moreover, since the training set of the model is a finite sample from an infinite set, a non robust solution is necessarily permitted. This is because any finite sample from $[-1, 1]^n$ will necessarily produce a region around the decision boundary (the standard bases of $\mathbb{R}^n$) from which no points are sampled; due to which multiple solutions are made possible of which only one is the true one. This is shown in Figure 4.1 where the yellow region around the axes represents the unsampled region. For example, if in the training data $|x_i| \geq \epsilon > 0$ then a valid solution, given this training data, might consider $0 < x_i < \epsilon$ to have sign -1, that is the decision boundary is drawn at the edge of the yellow region. If this happens then an adversary might be able to perturb x_i just enough that $0 < x_i < \epsilon$ and cause a mis-classification. One could indeed argue that the input with the perturbed x_i is not part of the training data distribution and is an example of an *off-distribution adversarial input*.

Based on the preceding discussion, we can define an adversarially robust model as a model that can make highly accurate predictions even when test data is not in the support of the training data distribution. Figure 4.1 illustrates an example of training and testing data distributions which differ as mentioned above. The yellow region is in the support of the test data but not of the training data. As a result the set of valid solutions i.e. those that achieve high accuracy on the training data but not necessarily on the test data, is a superset of the set of robust solutions, i.e. those that achieve high accuracy on the test data. If the prediction accuracy on the training data is the sole metric, the optimization procedure will not be able to identify the robust solution from amongst the valid solutions and hence will converge to the solution that is most accessible from the point of initialization. It follows that the probability with which a trained model will be adversarially robust is equal to the probability that a randomly initialized model is close to a robust solution. In the subsequent sections we first describe a monte carlo sampling based method for estimating the probability that a randomly initialized model will converge to a robust solution, then we present and analyse the estimates obtained by applying this method to various models trained to compute RXOR. In our analysis we aim to isolate trends that can be used to inform modeling choices that would facilitate the training of robust models.

4.2.1 Methodology

To estimate the probability of arriving at a robust solution we use a monte carlo sampling based approach, which is as follows. We begin by selecting an architecture for our neural network and initializing the parameters of the network by uniformly sampling a point from a p dimensional L2 norm ball, where p is the number of parameters of the network. The radius of the norm ball is chosen such that valid and robust solutions are possible within it. Next we use Stochastic Gradient Descent (SGD) to optimize the parameters to minimize the prediction error on a training dataset. In this step we perform only a small number of SGD updates because we only want to search for a solution in the vicinity of the random initialization. This is to ensure that we are visiting a wide range of solutions, in the case that SGD tends to end on a specific minimum given enough time. The distribution of the training data resembles Figure 4.1 in that points are sampled uniformly from $([-1,-\epsilon]\cup[\epsilon,1])^n$, where n is the number of inputs to RXOR and ϵ is the width of the unsampled region around the axes. Concretely, we sample data sets $\mathcal{X}_{\epsilon_1}...\mathcal{X}_{\epsilon_k}$ for $\epsilon_0 < ... < \epsilon_k$, such that $\mathcal{X}_{\epsilon_j} = \{\mathbf{s}\odot\mathbf{x}|\mathbf{x}\in\mathbb{R}^n, x_i \sim U[\epsilon_j,1], \mathbf{s}\in\{-1,1\}^n, P(s_i=-1)=P(s_i=+1)=0.5\}$, where $\odot$ represents element-wise multiplication. We then train the model on one of the data sets, say $\mathcal{X}_{\epsilon_j}$, and evaluate its accuracy on all of them. Note that the support of the test sets $\mathcal{X}_{\epsilon_{i<j}}$ includes a region that is not in the support of the training set. Based on the definition of robustness presented in the previous section, if a model trained on the training set is able to make perfect predictions on the testing set, we will consider it to be robust. More specifically, we will refer to a model that achieves 100% accuracy on $\mathcal{X}_{\epsilon_i}$ as being ϵ_i-robust. Of course, all models that are trained on $\mathcal{X}_{\epsilon_i}$ will be ϵ-robust for $\epsilon \geq \epsilon_i$ but only some of them might be robust for $\epsilon \leq \epsilon_i$.

For each value of ϵ_j, we repeat the above procedure N times to obtain N models that had different random initialization and have been trained and evaluated on different samplings of the training set $\mathcal{X}_{\epsilon_j}$ and $\mathcal{X}_{\epsilon_1}...\mathcal{X}_{\epsilon_k}$, respectively. Next we create a set of valid solutions for each ϵ_j, denoted by $\mathcal{V}_{\epsilon_j}$, containing models that were trained on $\mathcal{X}_{\epsilon_j}$ achieved 100% accuracy. We also create, for each ϵ_j, sets of robust solutions, $\mathcal{R}_{\epsilon_j,\epsilon_1},...,\mathcal{R}_{\epsilon_j,\epsilon_k}$, corresponding to different levels of robustness and containing models that achieve 100% accuracy on $\mathcal{X}_{\epsilon_1},...,\mathcal{X}_{\epsilon_k}$, respectively. Using these sets we estimate the following probabilities:

- $P(\text{robust}|\epsilon_i,\epsilon_j) \approx \frac{|\mathcal{R}_{\epsilon_j,\epsilon_i}|}{|\mathcal{V}_{\epsilon_j}|}$ – the probability of (quasi) randomly arriving at an ϵ_i-robust.
- $P(\text{valid}|j) \approx \frac{|\mathcal{V}_{\epsilon_j}|}{N}$ – the probability of arriving at a valid solution by training on $\mathcal{X}_{\epsilon_j}$
- $P^*(\text{robust}|\epsilon_i,\epsilon_j) \approx \frac{|\mathcal{V}_{\epsilon_i}|}{|\mathcal{V}_{\epsilon_j}|}$ – the oracle probability of robustness. This is the upper bound of $P(\text{robust}|\epsilon_i,\epsilon_j)$ that is achieved if $\mathcal{V}_{\epsilon_{i<j}} \subset \mathcal{V}_{\epsilon_j}$, i.e. all the solutions achievable by training on $\mathcal{X}_{\epsilon_i}$ are recovered by training on $\mathcal{X}_{\epsilon_j}$

4.2.2 Influence of hyperparameters on accurate and robust classifiers

Given an number m we define $F_{a_1,a_2,...,a_m}$ as the set of n-hidden-layer binary neural classifiers, that have a_1 neurons on the first hidden layer, a_2 on the second, etc. The final layer always has one neuron for binary classification. When there is no ambiguity on f, we name $h_{i,j}$ the j^{th} neuron of the i^{th} layer. We name $h'_{i,j}$ the pre-activation neuron (which is an affine function).

h_2	x, y	$-1,-1$	$-1,1$	$1,-1$	$1,1$
AND	$h_{1,1}$	0	1	1	1
	$h_{1,1}$	1	1	1	0
	$f(=\text{XOR})$	0	1	1	0
OR	$h_{1,1}$	0	1	0	0
	$h_{1,1}$	0	0	1	0
	$f = (\text{XOR})$	0	1	1	0

Table 4.1: The necessary truth tables of $h_{1,1}$ and $h_{1,2}$ when $h2$ is either an AND gate or an OR gate

We can establish the existence or non-existence of valid solutions on $\mathcal{X}_\epsilon$ for certain values of ϵ, n, m, a_i and certain activation functions.

Proposition 4.1. *For $n = 2$, with threshold activations, $F_{2,2}$ contains a classifier that is valid on $\mathcal{X}_\epsilon$ for all $0 < \epsilon < 1$.*

Proof. We exhibit an $f_r \in F_{2,2}$. We set:

- $h_{1,1} = (x_1 \geq 0)\ ,\ h_{1,2} = (x_2 \geq 0)$
- $h_{2,1} = h_{1,1} \wedge h_{1,2} = (h_{1,1} + h_{1,2} - 2 \geq 0)$
- $h_{2,2} = \neg h_{1,1} \wedge \neg h_{1,2} = (-h_{1,1} - h_{1,2} \geq 0)$
- $h_{3,1} = h_{2,1} \vee h_{2,2} = (h_{2,1} + h_{2,2} - 1 \geq 0)$

In other words h_1 turns the RXOR problem into the traditional BXOR problem, which the two-layer network h_2, h_3 can classically solve. Since h_1 sends (x_1, x_2) onto the associated vertex $(\pm 1, \pm 1)$, any point in $\mathcal{X}_\epsilon$ with $0 < \epsilon < 1$ is correctly classified by f_r.

□

It is obvious that any $F_{a_1,a_2,\ldots,a_m}$ with $m \geq 2$ and a_i also contain classifier f_r, since layers can easily ignore some neurons or compute the identity function. On the other hand, results change if we drop one layer.

Proposition 4.2. *For $n = 2$, with threshold activations, given $0 < \epsilon_1 < \frac{2}{3}$ and $\frac{2}{3} < \epsilon_2 < 1$, F_2 contains a valid classifier on $\mathcal{X}_{\epsilon_1}$ but not on $\mathcal{X}_{\epsilon_2}$*

Proof. In F_2 the same h_1 as above cannot lead to a valid classifier: the first layer must project the dataset onto a linearly separable dataset, which binary XOR isn't. In fact, the only non-trivial gates h_2 can modelize are AND ($a + b \geq 2$) or OR ($a + b \geq 1$), applied to the literals $(\neg)h_{1,1}$ and $(\neg)h_{1,2}$. Without loss of generality we can assume that $h_2 = h_{1,1} \vee h_{1,2}$ or $h_2 = h_{1,1} \wedge h_{1,2}$.

f must correctly classify vertices $(\pm 1, \pm 1)$. We can therefore reverse engineer the truth tables of $h_{1,1}$ and $h_{1,2}$ on each vertex, depending on what h_2 does.

- If h_2 is AND then we must have $h_{1,j}(-1,1) = h_{1,j}(1,-1) = 1$ for $j \in \{1,2\}$. For at least one j we have $h_{1,j}(-1,-1) = 0$: we assume it is $j = 1$. By affinity $h'_{1,1}(1,1) = h'_{1,1}(-1,1) + h'_{1,1}(1,-1) - h'_{1,1}(-1,-1) \geq 0$ as it is the sum of three positive terms: therefore $h_{1,1}(1,1) = 1$. For at least one j we have $h_{1,j}(1,1) = 0$: it must be is $j = 2$. Similarly we can infer that $h_{1,2}(-1,-1) = 1$. We write these truth tables in Table 4.1.

- If h_2 is OR, we can apply a similar reasoning and figure out the truth tables of $h_{1,1}$ and $h_{1,2}$, which we also report in Table 4.1.

We note that the $h'_{1,j}$ must keep a constant sign over each connected component of $\mathcal{X}_\epsilon$ (regions $[-1, \epsilon - 1] \times [-1, \epsilon - 1]$, $[-1, \epsilon - 1] \times [1 - \epsilon, 1]$, etc.), as any permissible change of activation for a single hidden neuron changes the final output. Respectively, if these conditions and the truth tables above are met for $h_{1,1}$ and $h_{1,2}$ then f is valid. In the h_2-OR case, writing

$$h_{1,1}(x, y) = ax + by \geq 1$$

the conditions on $h_{1,1}$ are equivalent to the following system of inequations:

$$\begin{cases} -a - (1 - \epsilon)b < 1 \\ (1 - \epsilon)a + b < 1 \\ -(1 - \epsilon)a + (1 - \epsilon)b \geq 1 \end{cases}$$

The existence of a solution depending on ϵ can be solved easily using linear programming. We find that the (a, b) satisfying the equality case for the first two inequations are $a == \frac{1}{\epsilon}, b = \frac{1}{\epsilon}$. On this corner point inequation 3 becomes

$$\frac{2 - 2\epsilon}{\epsilon} \geq 1$$

Therefore F_2 contains a valid solution on $\mathcal{X}_\epsilon$ iff $\epsilon < \frac{2}{3}$.

□

To an extent, a similar reasoning could be repeated with F_k for $k > 2$. However if that width k becomes extremely large, it is possible for the network to become arbitrarily close to f_r. This is because arbitrarily wide two-layer networks are universal approximators. Although we did not derive it, we expect that the maximal value of ϵ would slowly increase from $\frac{2}{3}$ to 1 when increasing the width.

These very wide networks aside, two layer networks are insufficient to achieve robust classification on XOR with threshold activations. This changes when picking the ReLU activation $\text{ReLU}(x) = x^+$:

Proposition 4.3. *For $n = 2$, with ReLU activations, F_3 contains a classifier that is valid on $\mathcal{X}_\epsilon$ for all $0 < \epsilon < 1$.*

Proof. It can be verified that with

$$h_{1,1} = x_1^+ \ , \ h_{1,2} = x_2^+ \ , \ h_{1,3} = (x_1 + x_2)^+$$

$$h_{2,1} = (h_{1,1} + h_{1,2} - h_{1,3} \geq 0)$$

then $f = f_r$ □

The key difference with threshold cases is the continuity of ReLU. It makes it possible for a first-layer pre-activation neuron to change sign within a connected component like $[-1, \epsilon - 1] \times [1 - \epsilon, 1]$, as $h'_{1,3}$ does, while having a valid classifier.

4.3 Experimental Setup

In our experiments we use Multi-Layered Perceptrons (MLP) without biases. We removed the bias parameter because it is not required to compute RXOR and removing it reduces the parameter count, which in turn reduces the memory and computation requirements of our experiments. We ran experiments with various model architectures in order to observe the impact of width, depth and activation functions on the probability of arriving at a robust solution. We set the radius of the L2 norm ball from which the initial values of the parameters are sampled to 25. We optimize the initial parameters by performing 20 updates using SGD with Nesterov momentum of 0.9.

To train and test the models we create datasets $\mathcal{X}_\epsilon$, as described in the previous section, for $\epsilon \in \mathcal{E} = \{0.4, 0.3, 0.2, 0.1, 0.05\}$ with each data set having 1000 points. We denote these 2D datasets as $\mathcal{X}^2_\epsilon$ for $\epsilon \in \mathcal{E}$. We train each model architecture on each of the datasets separately but we evaluate each trained model on all of the datasets. This allows us to determine what fraction of models trained on $\mathcal{X}^d_{0.4}$, also achieve 100% accuracy on $\mathcal{X}^d_{0.1}$. We will refer to such models as being ϵ-robust for $\epsilon = 0.1$. Of course, all models that are trained on $\mathcal{X}^d_{\epsilon_i}$ will be ϵ-robust for $\epsilon \geq \epsilon_i$ but only some of them might be robust for $\epsilon \leq \epsilon_i$.

To estimate the $P(\text{robust}|i, j)$ we perform $N = 25000$ trainings. In each training training and testing sets, and the initial parameters of the model are sampled randomly from their respective distributions.

4.4 Experimental Results

4.4.1 Model Architecture Notation

In this section we adopt the following notation to refer to the model architecture. We use "MLP-w_1_..._w_L-f_a" to refer to a MLP with L layers having widths $w_1, ..., w_L$ and activation function f_a. In the case dropout Srivastava et al. [2014] (with probability p) or batch norm Ioffe and Szegedy [2015] is used, the notation is changed to "MLP-w_1_..._w_L-f_aDropout$p$" or "MLP-w_1_..._w_L-f_aBN", respectively. If skip connections Srivastava et al. [2015] are introduced that connect layer i_1 to i_2, i_2 to i_3 and so on, then the notation becomes "MLP-w_1_..._w_L-f_a-wSkip_$i_1 > ... > i_k$". The notation used for CNNs is "CNN-$n_1 \times h_1 \times w_1 \times f_1 - s_1$_..._$w_L n_1 \times h_1 \times w_1 \times f_1 - s_1$-$f_a$" where h_i and w_i are the height and width of the convolutional kernel, f_i is the number of filters, s_i is the stride of the kernel and n_i is the number of consecutive layers that have the configuration $h_1 \times w_1 \times f_1 - s_1$. For example, Conv-1x7x7x32-3_2x7x7x16-3-ReLU represents a model with 3 layers, the first of which has a kernel of size 7x7, stride 3 and 32 filters while the remaining layers have the same kernel size and stride, but 16 filters instead of 32.

4.4.2 Probability of Robustness of Minimal Models

Here we present results from experiments conducted on *minimal* models i.e. having the number of parameters that are necessary and sufficient for solving RXOR. The minimal model for 2D inputs consists of 1 hidden layer with 3 ReLU units as discussed in section 4.2.2.

Figure 4.2 shows how $P(\text{robust}|\epsilon_i, \epsilon_j)$ and $P^*(\text{robust}|\epsilon_i, \epsilon_j)$ vary as the training and testing data margins, ϵ_j and ϵ_i respectively, are varied for 2D data. We see that for a fixed ϵ_j, both

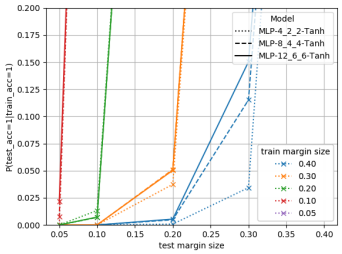


Figure 4.2: $P(\text{robust}|\epsilon_i, \epsilon_j)$ and $P^*(\text{robust}|\epsilon_i, \epsilon_j)$ curves for different values of ϵ_i (Test Data Margin) and ϵ_j(Train Data Margin) for the minimal model for 2D RXOR.

$P(\text{robust}|\epsilon_i, \epsilon_j)$ and $P^*(\text{robust}|\epsilon_i, \epsilon_j)$ decreases rapidly as ϵ_i is decreased to values less than ϵ_j. However, $P(\text{robust}|\epsilon_i, \epsilon_j)$ falls faster than $P^*(\text{robust}|\epsilon_i, \epsilon_j)$ and as a result as ϵ_i decreases, $\frac{\mathcal{V}_{\epsilon_i} \setminus \mathcal{V}_{\epsilon_j}}{\mathcal{V}_{\epsilon_i}}$ i.e. the proportion of valid solutions for $\mathcal{X}_{\epsilon_i}$ that are recovered by training on $\mathcal{X}_{\epsilon_j}$, also decreases. Stated another way, for a given ϵ_j, as ϵ_i is decreased not only does the number of *possible* ϵ_i-robust solution decreases but the ability of the training mechanism to actually find these solutions diminishes. This indicates that perhaps better training and sampling methods might improve the odds of arriving at ϵ_i-robust solution, while keeping ϵ_j fixed. On the other hand, we note that for a given ϵ_i both $P(\text{robust}|\epsilon_i, \epsilon_j)$ and $P^*(\text{robust}|\epsilon_i, \epsilon_j)$ increase as ϵ_j is decreased, which implies that regardless of the value of ϵ_i, reducing ϵ_j can improve the odds of finding an ϵ_i-robust model.

4.4.3 Influence of Modeling Choices on Probability of Robustness

In this section we present experimental results that show the influence different modeling choices have on $P(\text{robust}|\epsilon_i, \epsilon_j)$ and $P^*(\text{robust}|\epsilon_i, \epsilon_j)$. The modeling choices we consider here are the width of the network i.e. number of units in each layer, depth of the network i.e. the number of layers, the activation function, the dropout probability, is batch normalization used or not, and are skip connections used or not. To improve the clarity of presentation, in the remainder of this section we do not show the curves for $P(\text{robust}|\epsilon_i, \epsilon_j)$ and $P^*(\text{robust}|\epsilon_i, \epsilon_j)$ but, instead, we summarize the curve for each training margin (ϵ_j) by computing the area under it and presenting it as a bar chart in Figure **??**. For a given ϵ_j, we denote the area under the $P(\text{robust}|\epsilon_i, \epsilon_j)$ and $P^*(\text{robust}|\epsilon_i, \epsilon_j)$ curves as AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ respectively.

Width

Figure 4.3a shows AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ for models consisting of one hidden layer containing 3, 6 and 9 ReLU units and trained on 2D data. We note that as the width of the network is increased $P(\text{robust}|\epsilon_i, \epsilon_j)$ also increases, albeit at a decreasing rate. Nevertheless, the impact of increasing the number of units from 3 to 6 is substantial – AUC_{ϵ_j} almost double and $P(\text{robust}|0.05, 0.4)$ increases from almost 0% to 3% and $P(\text{robust}|0.1, 0.4)$ increases from 0.5% to 12%. These improvements take randomly achieving ϵ-robustness for small ϵ from being near impossible to being reasonably probable – one can expect to find a 0.9-robust model for every 9 models trained on $\mathcal{X}_{0.4}$.

Turning our attention to $\text{AUC}^*_{\epsilon_j}$, we note that $\text{AUC}^*_{\epsilon_j}$, and consequently $P^*(\text{robust}|\epsilon_i, \epsilon_j)$, increases very rapidly as width is increased. We also note that improvement in $P^*(\text{robust}|\epsilon_i, \epsilon_j)$

outstrips the improvement in $P(\text{robust}|\epsilon_i, \epsilon_j)$, and this gap only grows as the width of the network increases. This indicates that while wide models can find a large number of solutions when trained on $\mathcal{X}_{\epsilon_i}$ for small ϵ_i, they are unable to recover most of them when they are trained on $\mathcal{X}_{\epsilon_j > \epsilon_i}$.

Depth

Figure 4.3b shows AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ for four models, namely, MLP-6-ReLU, MLP-3_3-ReLU, MLP-9-ReLU and MLP-3_3_3-ReLU (see Section 4.4.1). Note that the first two and the last two models have the same number of parameters, 18 and 27 respectively, so differences in $P(\text{robust}|\epsilon_i, \epsilon_j)$ are attributable to a change in depth of the model. We observe that among models with the same number of parameters but different depths, the shallower models have higher AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ than the deeper models, indicating that both valid and robust solutions are a more difficult to find for the latter models. Furthermore, we note that the impact of increasing the number of parameters from 18 to 27 depends on whether the additional parameters increased the width of the network or its depth. As noted in Figure 4.3a and again in Figure $4.3b$, if the width is increased we see an appreciable increase in AUC_{ϵ_j}, however, if depth is increased the change in AUC_{ϵ_j} is hardly noticeable. Finally, we observe that increasing the depth of the model does cause $\text{AUC}^*_{\epsilon_j}$ to increase. This indicates that the additional parameters indeed enhance the ability of the model to find solutions for small testing margins, ϵ_i, but only if the training margin, ϵ_j, is close to ϵ_i.

The above experiments indicate that to improve the odds of finding robust solutions the width of the model must be increased and the depth decrease, however, if for some reason the depth can not be decreased one may ask the question which layer should be widened? If resources are plentiful then one may simply widen all the layers, but in case of resource paucity then one may want to know which layers, if widened, lead to the most improvement in the odds of achieving robustness. To investigate this question we use three layer models trained on 2D. The base model contains three ReLU units in each layer. We generate three wider models from the base model by increasing the width of each layer, one by one, by a factor of two. The AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ for these models data are shown in Figure 4.3c. We note that in most cases, increasing the width of the first layer seems to improve AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$.

Activation Function

To determine the impact of the activation function on the probability of arriving at robust solution we take the widest model architectures from 4.4.3, set the activation function to one of ReLU, Tanh or Sigmoid, and compute AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$. We also add biases to these models because for some of the activation functions the models can not learn RXOR without affine computations. Figure 4.3d shows that for all ϵ_j, AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ are maximal when ReLU activations are used.

Batch Normalization

To observe the effect of batch normalization, we take MLP-9-ReLU and add a batch normalization layer after the linear layer and before the ReLU to obtain MLP-9-ReLUBN. Figure 4.3e shows AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ for the two models. We observe that the addition of batch normalization has reduced both AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ meaning that the introduction of batch normalization has made both robust and valid solutions less accessible. This observation conflicts with our expectation

since batch normalization is known to improve accuracy if training and testing data are identically distributed, but, perhaps this effect is due to simplistic nature of the RXOR problem. However, it is in line with our expectation and existing literature **?** that batch normalization reduces the likelihood of finding a robust solution.

Dropout

To observe the effect of dropout, we take MLP-9-ReLU and apply dropout with probability p after the linear layer and before the ReLU to obtain MLP-9-ReLUDropout$p$, where $p \in \{0.3, 0.6\}$. From Figure 4.3f we see that the addition of dropout with $p = 0.3$ increases AUC_{ϵ_j} but reduces $\text{AUC}^*_{\epsilon_j}$. This indicates that regularization provided dropout prevents the training algorithm from fitting the decision boundary too close to the training data. Increasing p to 0.6, however, leads to a reduction in both AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$, which indicates that this level of regularization diminishes the model's capacity to model the data.

Skip Connections

To test the impact of skip connections we take a two layer model with 9 ReLU units in each layer (MLP-9_9-ReLU), and add a skip connection from the output of the first layer to the output of the second layer (MLP-9_9-ReLU-wSkip_1>2). Our expectation is that the addition of skip connections would increase the likelihood of finding robust models compared to MLP-9_9-ReLU because the training algorithm can set the weights and biases in the second layer to zero and recover a single layer model with 9 hidden units, which, based on the above results, should be more robust than a two layer model. Looking at Figure 4.3g we note that it is indeed the case that skip connections improve AUC_{ϵ_j} and, to a lesser extent, $\text{AUC}^*_{\epsilon_j}$.

To summarize, the experimental results presented above have yielded the following insights: (1) wider models are likely to be more robust than narrower models, (2) deeper models are likely to be less robust than shallower models, (3) increasing the width of the earlier layers improves the odds of robustness, and (4) the activation function that maximizes the odds of robustness depends on the function being modeled. These insights together suggest that the odds of robustness may be improved by first finding the minimal architecture for the task, ideally in terms of the number of parameters since that may involve selecting the best activation function but if that is not possible than in terms of the number of layers, and then widening this architecture as much as possible.

4.5 Generalization to More Complex Data and Models

In Section 4.4 we considered simple models (MLPs) trained on a very simple dataset (2D RXOR), however, the datasets and models used in practice are much more complex. If the general trends relating different modeling choices to the robustness potential that are observed on the simple models and datasets generalize to more complex ones, then we can use them as rules-of-thumb when training models for practical tasks. In order to verify if the trends from 4.4 hold when more complex datasets and models are used, we repeat the experiments using larger MLPs and Convolutional Neural Networks (CNNs) trained on the MNIST dataset LeCun et al. [2010]. The methodology for the experiments is similar to the one presented in 4.2.1 with the following changes:

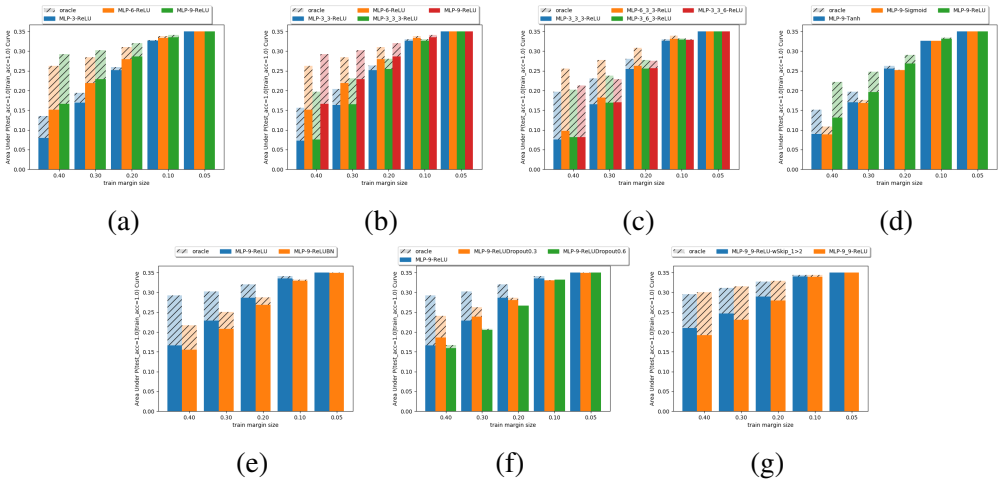


Figure 4.3: The influence of various modeling choices on $P(\text{robust}|\epsilon_i, \epsilon_j)$ as measured by AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ for MLPs trained on 2D RXOR data. Subfigure (a) shows the areas for single layer MLPs with increasing width, (b) shows the areas for MLPs with increasing depth, (c) shows the effect of widening different layers in a 3 layer MLP, (d) shows the effect of changing the activation function, (e) shows the impact of adding batch normalization, (f) shows the impact of adding dropout and (g) shows the impact of adding skip connections in a 2 layer MLP.

1. The notion of the margin used in 4.2.1 is not applicable to the natural image classification task and needs to be modified because, unlike the RXOR problem, the decision boundaries are unknown. Therefore, instead of considering a margin of width ϵ around the decision boundary beyond which all data lies we consider a margin around the datapoint. This corresponds to a hypercube having sides of length 2ϵ to be specific, around each input in the dataset from which we sample training and testing datapoints. To sample hypercube efficiently we run Projected Gradient Descent (PGD) Madry et al. [2018a] for a fixed number of steps to find data points within the 2ϵ hypercube that are not correctly classified by the model. This change necessitates redefining the data sets as $\mathcal{X}_\epsilon := \{x + \delta | \delta = \arg\max_{\delta:\|\delta\|_\infty \leq \epsilon} \mathcal{L}(f(x+\delta), y)\}$. where x is an image from the dataset, y is the true label of the image, f is the model and $\mathcal{L}$ is a loss function that indicates how close the prediction, $f(x+\delta)$ is to y. If no point is found within the hypercube for which the model makes an erroneous prediction we conclude that the model is ϵ-robust. Due to this change in the remainder of this section we use the training margin, ϵ_j, and testing margin, ϵ_i, to refer to *the margin around the data point and not around the decision boundary*.

2. In practice, however, the condition that every point in the hypercube must be classified correctly is unrealistic because currently 100% accuracy is not achievable on these datasets even for $\epsilon = 0$. Therefore, We relax the definitions of valid and robust solutions such that a valid solution is one that achieves accuracy at least τ_v on the training data, and a robust solution is one that achieves accuracy at least τ_r on the testing data. All the experimental results that follow have been obtained with $\tau_v = \tau_r = 0.85$.

3. We increase the number of SGD updates performed on randomly sampled parameters. This is done because the natural image classification task is more complex, training accurate

models requires more iterations.

4.5.1 Experimental Results

Trends in MLPs

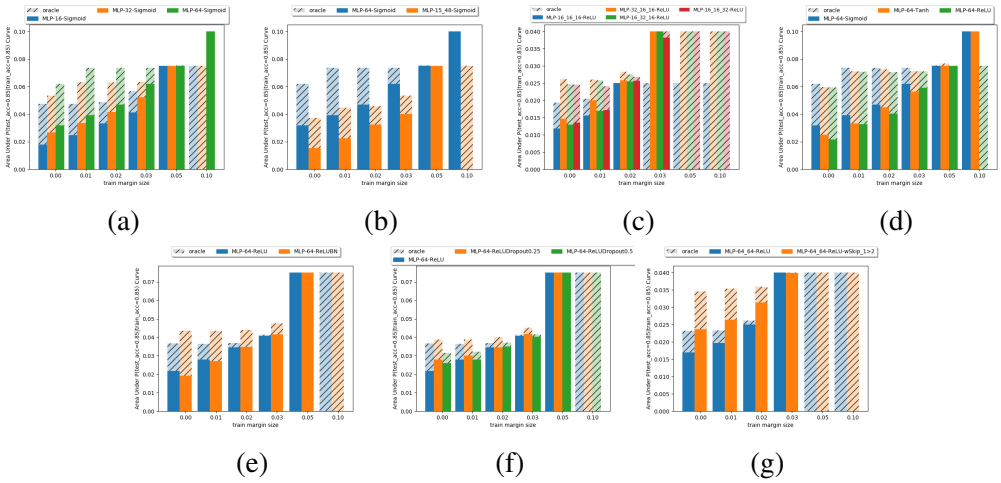
While MLPs are generally not used in computer vision tasks, we include them in our evaluation in order to maintain a degree of similarity with earlier experiments that would allow us to gauge the impact of increased the data complexity and dimensionality on the robustness potential of the models.

Figure 4.4 shows the AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ for different modelling choices. We see that the relationship between different modeling choices and, AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$, that we observed in models trained on 2D RXOR has carried over to models trained on the much more complex MNIST data. Specifically, we see that the likelihood of finding a solution that generalizes to smaller ϵ_i, i.e. an ϵ_i-robust solution for $\epsilon_i > \epsilon_j$, is increased by increasing the width of the model, particularly of the earlier layers, applying dropout with appropriate probability and introducing skip connections. On the other hand, increasing the depth of the network and applying batch normalization reduces the likelihood of finding an ϵ_i-robust solution.

While the general trends remain consistent between models trained on 2D RXOR and MNIST, there are some differences that are worth noting. Firstly, in the case of 2D RXOR models that used the ReLU activation performed better (see Section 4.4.3) whereas we see from Figure 4.4d that MLPs with Sigmoid units improve the robustness potential of the models trained on MNIST. This observation prevents us from making any claims about the relationship between robustness potential and activation function, rather we posit that certain activation functions are better suited to certain models and dataset. Secondly, we note from Figure 4.4e that adding batch normalization to a model increases $\text{AUC}^*_{\epsilon_j}$ but decreases AUC_{ϵ_j}. This observation is consistent with our expectation and existing literature since it is common to use batch normalization to increase the classification accuracy Ioffe and Szegedy [2015] and a recent study **?** has shown that batch normalization is detrimental to adversarial robustness because the normalization parameters are ill suited for perturbed data.

Trends in Convolutional Neural Networks

Having verified that the relationship between modeling choices and robustness potential transcends data complexity, we run experiments to determine if this relationship is maintained when the complexity of the model is increased. To this end, instead of using MLPs, the experimental results presented in this section use CNNs. At each layer of a CNN the output is computed by diving the input (or the output of the previous layer) into segments and applying a single layer MLP to each input segment. Therefore the modeling choices that apply to MLPs also apply to CNNs. Note that the number of convolutional filters represents the width of the model. In addition to the modeling choices common between CNNs and MLPs, there are some CNN specific modeling choices related to how the input is to be segmented. In practice segmentation is performed via sliding a window over the spatio-temporal dimensions of input and the modeler may choose the size of the window and the *stride* by which the window is moved in each step. Based on the sizes of the window



(a) (b) (c) (d)

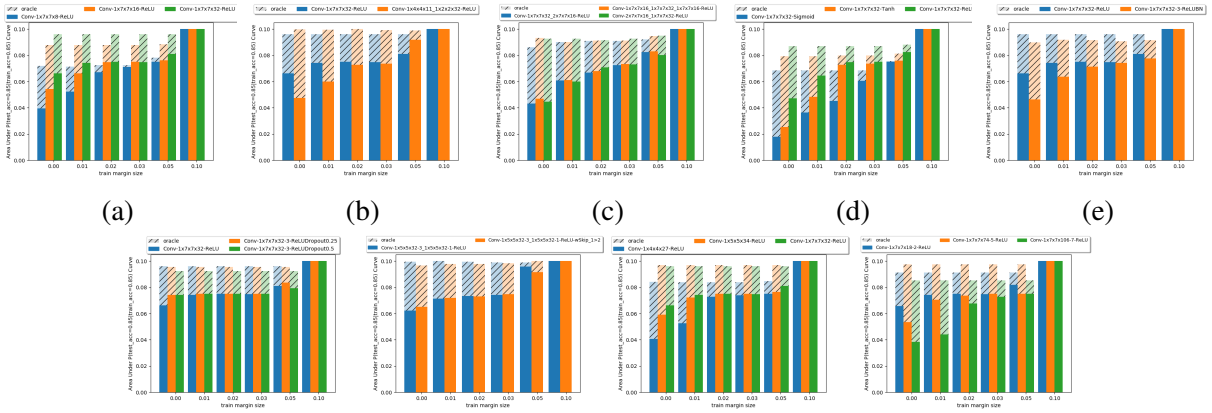
(e) (f) (g)

Figure 4.4: The influence of various modeling choices on $P(\text{robust}|\epsilon_i, \epsilon_j)$ as measured by AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ for MLPs trained on MNIST. Subfigure (a) shows the areas for single layer MLPs with increasing width, (b) shows the areas for MLPs with increasing depth, (c) shows the effect of widening different layers in a 3 layer MLP, (d) shows the effect of changing the activation function, (e) shows the impact of adding batch normalization, (f) shows the impact of adding dropout and (g) shows the impact of adding skip connections in a 2 layer MLP.

and stride in the previous layers, we can determine for a layer its *receptive field*, i.e. the effective window size for the layer with respect to the input to the model.

Figure 4.5 shows the AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ for different modelling choices. Considering the choices that are common between MLPs and CNNs first, we note that the general trend observed in Sections 4.4 and 4.5.1 are present here as well with one exception that is increasing the width of the middle layer and the last layer yields higher AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ than increasing the width of the first layer (see Figure 4.5c), with the middle layer yielding the highest AUC_{ϵ_j}. We hypothesize that this difference arises because unlike the MLP, the CNN processes segments of the input. The receptive field of the CNN increases at deeper layers so the additional width is more useful when the model is processing a larger part of the input.

Turning our attention to modeling choices specific to CNNs, namely the size of the receptive field (Figure 4.5h) and the stride (Figure 4.5i), we note that models that have a larger receptive fields and smaller strides tend to have greater robustness potential than those with smaller receptive field and larger strides, respectively. Changing the receptive field and the stride can change the total number of model parameter so we slightly modified the width of the network such that the total number of parameters in the models being compared remained similar. This change of width does not confound the results the widest model is not the one with the highest AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ in Figures 4.5h and 4.5i.



(a) (b) (c) (d) (e)

(f) (g) (h) (i)

Figure 4.5: The influence of various modeling choices on $P(\text{robust}|\epsilon_i, \epsilon_j)$ as measured by AUC_{ϵ_j} and $\text{AUC}^*_{\epsilon_j}$ for CNNs trained on MNIST. Subfigure (a) shows the areas for single layer CNN with increasing number of filter, (b) shows the areas for CNNs with increasing depth, (c) shows the effect of widening different layers in a 3 layer CNN, (d) shows the effect of changing the activation function, (e) shows the impact of adding batch normalization, (f) shows the impact of adding dropout, (g) shows the impact of adding skip connections in a 2 layer CNN, (H) shows the impact of increasing the size of the convolutional kernel, and (I) shows the impact of increasing the stride of the convolutional kernel.

Part II

Biological Robustness Priors

Chapter 5

Training on Foveated Images Improves Robustness to Adversarial Attacks

5.1 Problem and Motivation

Deep Neural Networks (DNNs) are exceptionally adept at many computer vision tasks and have emerged as one of the best models of the biological neurons involved in visual object recognition Yamins et al. [2014], Cadieu et al. [2014]. However, their lack of robustness to subtle image perturbations that humans are largely invariant Szegedy et al. [2014], Geirhos et al. [2018b], Dodge and Karam [2017] to has raised questions about their reliability in real-world scenarios. Of these perturbations, perhaps the most alarming are *adversarial attacks*, which are specially crafted distortions that can change the response of DNNs when added to their inputs Szegedy et al. [2014], Ilyas et al. [2019] but are either imperceptible to humans or perceptually irrelevant enough to be ignored by them.

While several defenses have been proposed over the years to defend DNNs against adversarial attacks, only a few of them have sought inspiration from biological perception, which, perhaps axiomatically, is one of the most robust perceptual systems in existence. Instead, most methods seek to *teach* DNNs to be robust to adversarial attacks by exposing them to adversarially perturbed images Madry et al. [2018b], Wong et al. [2019a], Zhang et al. [2019] or random noise Cohen et al. [2019a], Fischer et al. [2020], Carlini et al. [2022] during training. While this approach is highly effective in making DNNs robust to the types of perturbations used during training, the robustness often does not generalize to other types of perturbations Joos et al. [2022], Sharma and Chen [2017], Schott et al. [2018]. In contrast, biologically-inspired defenses seek to make DNNs robust by integrating into them biological mechanisms that would bring their behavior more in line with human/animal vision Paiton et al. [2020], Bai et al. [2021], Dapello et al. [2020], Jonnalagadda et al. [2022], Luo et al. [2015], Gant et al. [2021], Vuyyuru et al. [2020]. As these defenses do not require DNNs to be trained on any particular type of perturbation, they yield models that, like humans, are robust to a variety of perturbations Dapello et al. [2020] in addition to adversarial attacks. For this reason, and in light of the evidence indicating a positive correlation between biological alignment and adversarial robustness Dapello et al. [2020], Harrington and Deza [2021], we believe biologically inspired defenses are more promising in the long run.

Following this line of inquiry, we investigate the contribution of low-fidelity visual sensing that

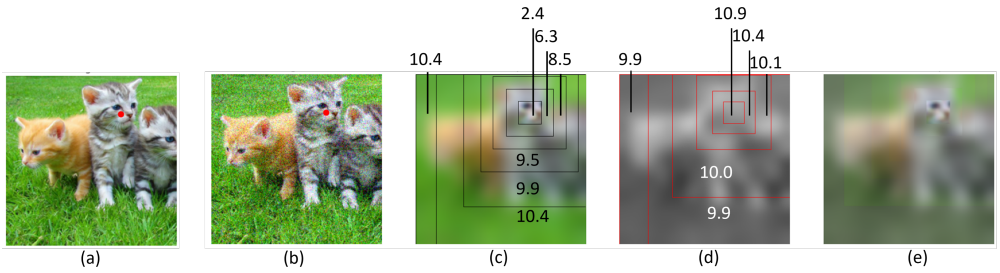


Figure 5.1: *R-Blur* adds Gaussian noise to image (a) with the fixation point (red dot) to obtain (b). It then creates a colored and a grayscaled copy of the image and applies adaptive Gaussian blurring to them to obtain the low-fidelity images (c) and (d), where the numbers indicate the standard deviation of the Gaussian kernel applied in the region bounded by the boxes. The blurred color and gray images are combined in a pixel-wise weighted combination to obtain the final image (e), where the weights of the colored and gray pixels are a function of their respective estimated acuity values (see 5.2.2).

occurs in peripheral vision to the robustness of human/animal vision. Unlike DNNs, which sense visual stimuli at maximum fidelity at every point in their visual field, humans sense most of their visual field in low fidelity, i.e without fine-grained contrast Stewart et al. [2020] and color information Hansen et al. [2009]. In adults with fully developed vision, only a small region (less than 1% by area) of the visual field around the point of fixation Kolb [2005] can be sensed with high fidelity. In the remainder of the visual field (the periphery), the fidelity of the sensed stimuli decreases exponentially with distance from the fixation point Dragoi and Tsuchitani [2020]. This phenomenon is called "foveation". Despite this limitation, humans can accurately categorize objects that appear in the visual periphery into high-level classes Ramezani et al. [2019]. Meanwhile, the presence of a small amount of noise or blurring can decimate the accuracy of an otherwise accurate DNN. Therefore, we hypothesize that the experience of viewing the world at multiple levels of fidelity, perhaps even at the same instant, causes human vision to be invariant to low-level features, such as textures, and high-frequency patterns, that can be exploited by adversarial attacks.

In this thesis, we propose *R-Blur* (short for Retina Blur), which simulates foveation by blurring the image and reducing its color saturation adaptively based on the distance from a given fixation point. This causes regions further away from the fixation point to appear more blurry and less vividly colored than those closer to it. Similar to how the retina preprocesses the visual stimuli before it reaches the visual cortex, we use *R-Blur* to preprocess the input before it reaches the DNN.

5.2 *R-Blur* Overview

To simulate the loss in contrast and color sensitivity of human perception with increasing eccentricity, we propose *R-Blur*, an adaptive Gaussian blurring, and color desaturation technique. The operations performed by *R-Blur*, given an image and fixation point, are shown in Figure 5.1. First, *R-Blur* adds Gaussian noise to the image to simulate stochastic firing rates of biological photoreceptors Croner et al. [1993]. It then creates color and grayscale copies of the image and estimates the acuity of color and grayscale vision at each pixel location, using distributions that approximate

the relationship between distance from the fixation point (eccentricity) and visual acuity levels in humans. ***R-Blur*** then applies *adaptive* Gaussian blurring to both image copies such that the standard deviation of the Gaussian kernel at each pixel in the color and the grayscale image is a function of the estimated color and grayscale acuity at that pixel. Finally, ***R-Blur*** combines the two blurred images in a pixel-wise weighted combination in which the weights of the colored and gray pixels are a function of their respective estimated acuity values. Below we describe some of the more involved operations in detail.

5.2.1 Eccentricity Computation

The distance of a pixel location from the fixation point, i.e. its eccentricity, determines the standard deviation of the Gaussian kernel applied to it and the combination weight of the color and gray images at this location. While eccentricity is typically measured radially, in this paper we use a different distance metric that produces un-rotated square level sets. This property allows us to efficiently extract regions having the same eccentricity by simply slicing the image tensor. Concretely, we compute the eccentricity of the pixel at location (x_p, y_p) as

$$e_{x_p,y_p} = \frac{\max(|x_p - x_f|, |y_p - y_f|)}{W_V}, \tag{5.1}$$

where (x_f, y_f) and W_V represent the fixation point and the width of the visual field, i.e. the rectangular region over which ***R-Blur*** operates and defines the maximum image size that is expected by ***R-Blur***. We normalize by W_V to make the e_{x_p,y_p} invariant to the size of the visual field.

5.2.2 Visual Acuity Estimation

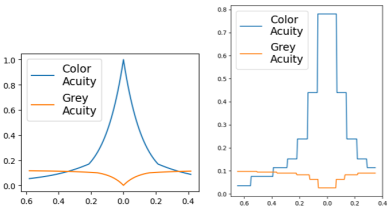
We compute the visual acuity at each pixel location based on its eccentricity. The biological retina contains two types of photoreceptors. The first type, called cones, are color sensitive and give rise to high-fidelity visual perception at the fovea, while the second type, called rods, are sensitive to only illumination but not color and give rise to low-fidelity vision in the periphery. We devise the following two sampling distributions, $D_R(e_{x,y})$ and $D_C(e_{x,y})$, to model the acuity of color and grayscale vision, arising from the cones and rods at each pixel location, (x, y).

$$\mathcal{D}(e; \sigma, \alpha) = \max\left[\lambda(e; 0, \sigma), \gamma(e; 0, \alpha\sigma)\right] \tag{5.2}$$

$$D_C(e; \sigma_C, \alpha) = \mathcal{D}(e; \sigma_C, \alpha) \tag{5.3}$$

$$D_R(e; \sigma_R, \alpha, p_{max}) = p_{max}(1 - \mathcal{D}(e; \sigma_R, \alpha)), \tag{5.4}$$

where $\lambda(.; \mu, \sigma)$ and $\gamma(.; \mu, \sigma)$ are the PDFs of the Laplace and Cauchy distribution with location and scale parameters μ and σ, and α is a parameter used to control the width of the distribution. We set $\sigma_C = 0.12, \sigma_R = 0.09, \alpha = 2.5$ and $p_{max} = 0.12$. We choose the above equations and their parameters to approximate the curves of photopic and scotopic visual acuity from Dragoi and Tsuchitani [2020]. The resulting acuity estimates are shown in Figure 5.2b. Unfortunately, the measured photopic and scotopic acuity curves from Dragoi and Tsuchitani [2020] cannot be reproduced here due to copyright reasons, however, they can be viewed at https://nba.uth.tmc.edu/neuroscience/m/s2/chapter14.html (see Figure 14.3).



(a) unquantized (b) quantized

Figure 5.2: Estimated visual acuity of sharp and colorful, photopic, and gray and blurry, scotopic, vision using equations 5.3 and 5.4

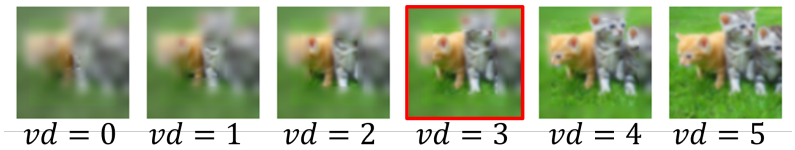


Figure 5.3: Illustration of increasing the viewing distance (left to right). As the viewing distance is increased, more of the image is brought into focus. We used $vd = 3$ during inference.

5.2.3 Quantizing the Visual Acuity Estimate

In the form stated above, we would need to create and apply as many Gaussian kernels as the distance between the fixation point and the farthest vertex of the visual field. This number can be quite large as the size of the image increases and will drastically increase the per-image computation time. To mitigate this issue we quantize the estimated acuity values. As a result, the locations to which the same kernel is applied no longer constitute a single pixel perimeter but become a much wider region (see Figure 5.1 (c) and (d)), which allows us to apply the Gaussian kernel in these regions very efficiently using optimized implementations of the convolution operator.

To create a quantized eccentricity-acuity mapping, we do the following. We first list all the color and gray acuity values possible in the visual field by assuming a fixation point at $(0, 0)$, computing eccentricity values $e_{0,y}$ for $y \in [0, W_V]$ and the corresponding values of $\mathcal{D}_R = \{D_R(e_{0,y})|y \in [0, W_V]\}$ and $\mathcal{D}_C = \{D_C(e_{0,y})|y \in [0, W_V]\}$. We then compute and store the histograms, H_R and H_C, from $\mathcal{D}_R$ and $\mathcal{D}_C$, respectively. To further reduce the number of kernels we need to apply and increase the size of the region each of them is applied to, we merge the bins containing less than τ elements in each histogram with the adjacent bin to their left. After that, given an image to process, we will compute the color and gray visual acuity for each pixel, determine in which bin it falls in H_R and H_C, and assign it the average value of that bin.

5.2.4 Changing the Viewing Distance

Increasing the viewing distance can be beneficial as it allows the viewer to gather a more global view of the visual scene and facilitates object recognition. To increase the viewing distance we drop the k lowest acuity bins and shift the pixels assigned to them k bins ahead such that the pixels that were in bins 1 through $k - 1$ are now assigned to bin 1. Figure 5.3 shows the change in the viewing distance as the value of k increases from 0 to 5. Formally, given the quantized $D_C(e_{x,y})$ and $D_R(e_{x,y})$, let $D = [d_1, ..., d_n]$ represent the value assigned to each bin and P_i be the pixel locations assigned to the i^{th} bin, with P_1 and P_n corresponding to points with the lowest and highest eccentricity, respectively. To increase the viewing distance, we merge bins 1 through k such that $D' = [d_1, ..., d_{n-k}]$ and the corresponding pixels are $P'_1 = [P_1, ..., P_k]$ and $P_{i>1} = P_{k+1}$.

5.2.5 Blurring and Color Desaturation

We map the estimated visual acuity at each pixel location, (x_p, y_p), to the standard deviation of the Gaussian kernel that will be applied at that location as $\sigma_{(x_p,y_p)} = \beta W_V(1 - D(e_{x,y}))$, where β is constant to control the standard deviation and is set to $\beta = 0.05$ in this paper, and $D = D_C$ for pixels in the colored image and $D = D_R$ for pixels in the grayscaled image. We then apply Gaussian kernels of the corresponding standard deviation to each pixel in the colored and grayscale image to obtain an adaptively blurred copy of each, which we combine in a pixel-wise weighted combination to obtain the final image. The weight of each colored and gray pixel is given by the normalized color and gray acuity, respectively, at that pixel. Formally, the pixel at (x_p, y_p) in the final image has value

$$v^f_{(x_p,y_p)} = \frac{v^c_{(x_p,y_p)} D_C(e_{x,y}; \sigma_C, \alpha) + v^g_{(x_p,y_p)} D_R(e_{x,y}; \sigma_C, \alpha)}{D_C(e_{x,y}; \sigma_C, \alpha) + D_R(e_{x,y}; \sigma_C, \alpha)}, \tag{5.5}$$

$v^c_{(x_p,y_p)}$ and $v^g_{(x_p,y_p)}$ are the pixel value at (x_p, y_p) in the blurred color and gray images respectively.

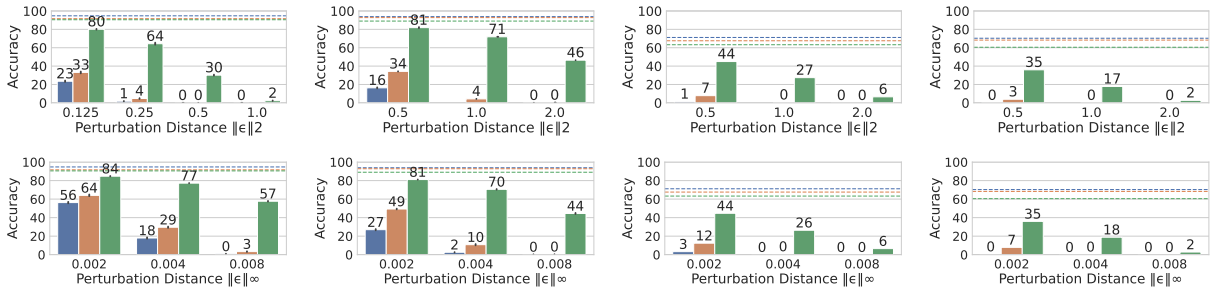
5.3 Key Results

Datasets: We use natural image datasets, namely CIFAR-10 Krizhevsky et al., Imagenet ILSVRC 2012 Russakovsky et al. [2015], Ecoset Mehrer et al. [2021] and a 10-class subset of Ecoset (Ecoset-10). Ecoset contains around 1.4M images, mostly obtained from ImageNet database Deng et al. [2009] (not the ILSVRC dataset), that are organized into 565 basic object classes. The classes in Ecoset correspond to commonly used nouns that refer to concrete objects. To create Ecoset-10, we select 10 classes from Ecoset that have the highest number of images. The training/validation/test splits of Ecoset-10 and Ecoset are 48K/859/1K, and 1.4M/28K/28K respectively. For most experiments with Ecoset and Imagenet, we use 1130, and 2000 test images, with an equal number of images per class. During training, we use random horizontal flipping and padding + random cropping, as well as AutoAugment Cubuk et al. [2018] for CIFAR-10 and RandAugment for Ecoset and Imagenet. All Ecoset and Imagenet images were resized and cropped to 224×224. We applied these augmentations to *all* the models we trained – those with biological and non-biological defenses, as well as the baseline models.

Model Architectures: For CIFAR-10 we use a Wide-Resnet Zagoruyko and Komodakis [2016] model with 22 convolutional layers and a widening factor of 4, and for Ecoset and Imagenet we use XResNet-18 from fastai Howard and Gugger [2020] with a widening factor of 2. Moving forward, we will refer to both these models as ResNet and indicate only the training/evaluation datasets from which the exact architecture may be inferred.

***R-Blur* improves robustness to white-box attacks.** We evaluate robustness by measuring the accuracy of models under Auto-PGD (APGD)Croce and Hein [2020c] attack, which is a state-of-the-art white-box adversarial attack. To determine if *R-Blur* improves robustness, we compare *R-Blur* to two baselines under the APGD attack: (1) an unmodified ResNet trained on clean data (ResNet), and (2) a ResNet which applies five affine transformations [1] to the input image and

[1]We apply rotation, translation, and shearing, with their parameters sampled from $[-8.6°, 8.6°]$, $[-49, 49]$ and



(a) CIFAR-10 (b) Ecoset-10 (c) Ecoset (d) Imagenet

Figure 5.4: Comparison of accuracy on various datasets (a-d) under adversarial attacks of several ℓ_2 (top) and ℓ_∞ (bottom) norms between *R-Blur* (green) and two baseline methods: *RandAffine* (orange) and ResNet (blue). The dashed lines indicate accuracy on clean images. *R-Blur* models consistently achieve higher accuracy than baseline methods on all datasets, and adversarial perturbation sizes.

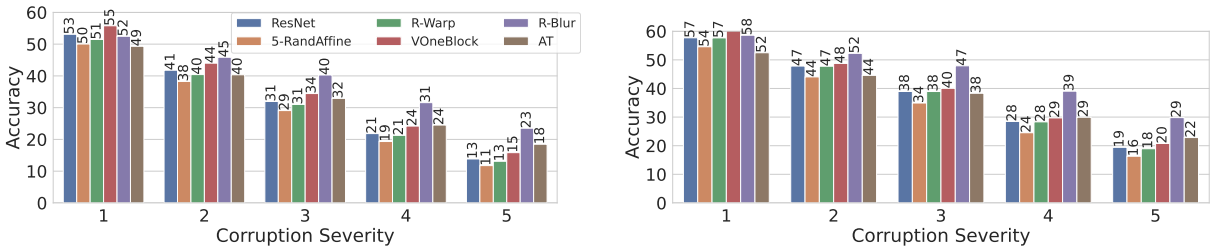
averages the logits (*RandAffine*). We observe that *R-Blur* is significantly more robust than the unmodified ResNet and *RandAffine* models, consistently achieving higher accuracy than the two on all datasets and against all perturbation types and sizes, while largely retaining accuracy on clean data (Figure 5.4). Particularly, on larger datasets – Ecoset and Imagenet, even the smallest amount of adversarial perturbation ($\|\delta\|_\infty = 0.002$, $\|\delta\|_2 = 0.5$) is enough to drive the accuracy of the baselines to $\sim$10%, while *R-Blur* still is able to achieve 35-44% accuracy. As the perturbation is increased to $\|\delta\|_\infty = 0.004$ and $\|\delta\|_2 = 1.0$, the accuracy of the baselines goes to 0%, while *R-Blur* achieves 18-22%.

***R-Blur* improves accuracy on common (non-adversarial) corruptions.** Adversarial perturbations constitute only a small subset of perturbations that human vision is invariant to, therefore we evaluate *R-Blur* on a set of common image corruptions Hendrycks and Dietterich [2019] that humans are largely invariant to but DNNs are not. We sample 2 images/class from Imagenet and 5 images/class from Ecoset. Then we apply 17 [2] common corruptions proposed in Hendrycks and Dietterich [2019] at 5 different severity levels to generate 85 corrupted versions of each image. This yields corrupted versions of Imagenet and Ecoset containing 170K and 240K images, respectively.

Figure 5.5 shows the accuracy of the models on corrupted Ecoset and Imagenet. Here we also compare against an adversarially trained model (*AT*) trained with $\|\delta\|_\infty = 0.008$ using the method of Wong et al. [2019a]. We see that at severity greater than 1 *R-Blur* consistently achieves the highest accuracy. Furthermore, we also note that *R-Blur*, and *VOneBlock* consistently achieve higher accuracy than *AT*, which supports our hypothesis that the robustness of biologically motivated methods, and particularly *R-Blur*, is more general than non-biological defenses, like *AT*. In fact, the accuracy of *AT* on common corruptions is generally lesser than or at par with the accuracy of the unmodified ResNet, indicating that the robustness of *AT* does not generalize well.

$[-8.6°, 8.6°]$ respectively. The ranges are chosen to match the ranges used in RandAugment. The random seed is fixed during evaluation to prevent interference with adversarial attack generation.

[2] We exclude Gaussian blur and Gaussian noise since they are similar to the transformations done by *R-Blur*.



(a) Imagenet

(b) Ecoset

Figure 5.5: The accuracy of the models on Imagenet and Ecoset under the common corruptions from Hendrycks and Dietterich [2019] at various severity levels. We see that *R-Blur* generally achieves the highest accuracy.

***R-Blur* alignment with human perception.** We use the *metamer test* proposed by Feather et al. [2019] to determine the extent to which *R-Blur* and standard ResNet models are aligned with human perception. Specifically, the metamer test seeks to determine whether the latent representations of a DNN are invariant to the same visual features that human visual perception is invariant to. Metamers are stimuli that appear indistinguishable to humans under certain conditions. In the context of this test, metamers are defined as inputs that induce the same (or similar) latent representations in the DNN.

To conduct the metamer test, Feather et al. [2019] generate metamers for a set of images and ask human subjects to classify the metamers. If the humans are able to correctly classify the metamers, the types of transformations that the model's latent representations are invariant to are similar to those that human perception is invariant to as well, and thus the model is a good approximation of human perception. The metamers are generated by using gradient descent to optimize a randomly initialized input tensor such that it induces the same latent representation(s) in the DNN as a given natural image. Formally, the following optimization problem is solved to obtain the metamer, $m_\phi(x)$, for a given image, x and a DNN, ϕ:

$$m_\phi(x) = \arg\min_m \|f_\phi(x) - f_\phi(m)\|_2 \,, \tag{5.6}$$

where f_ϕ represents the function that maps an input image to the latent representation of a DNN, ϕ.

Chapter 6

Fixed Inter-Neuron Covariability Induces Adversarial Robustness

6.1 Problem and Motivation

As discussed before, integrating mechanisms of biological perception into DNNs, improve their robustness. An aspect of human perception that is not well represented in DNNs is the inflexible inter-neuron covariability structure of biological neurons. It has been observed that the spiking activity of biological neurons tends to be correlated Hennig et al. [2021], Sadtler et al. [2014] and, the structure of this correlation tends to persist over long periods of time even if it limits performance and learning Golub et al. [2018]. In contrast, the activations of DNN neurons are not constrained in this way, making it possible to induce arbitrary activation patterns by adding perturbations that activate only specific neurons Paiton et al. [2020]. This allows an adversary to induce activation patterns that lead to misclassifications. Indeed, our experiments in this paper show that adversarial perturbations cause the inter-neuron correlations to change significantly. Therefore, we hypothesize that constraining the inter-neuron correlation in DNNs may improve their robustness. Integrating this constraint into DNNs is not straightforward, because, unlike biological neurons, neurons in a DNN do not produce stochastic spike trains, rather they output a deterministic real number. So, how can we simulate correlated spiking in a system that is neither stochastic nor produces spiking activity? To solve this issue, we consider the outputs of the artificial neuron as the frequency of an underlying spike train Hennig et al. [2021]. If the spiking activity of a group of neurons is correlated, the frequency of their spikes may also be correlated. Therefore, we use spiking frequency as a proxy for the spikes trains. Since the spiking frequency is represented in a DNN by the outputs of the neurons, we impose a fixed covariability structure on the latter.

To simulate a fixed inter-neuron covariability pattern, we develop the Self-Consistent Activation (SCA) layer, which comprises of neurons whose activations are consistent with each other as they conform to a fixed covariability pattern. The SCA layer first computes the feed-forward activations for the neurons based on the input, and then iteratively optimizes these activations to make them conform to a fixed, but learned covariability pattern.

6.2 Covariability of DNN Activations

We hypothesize that the inflexible covariability structure of neuronal activations that is observed in the animal brain contributes to the robustness of biological vision. As a preliminary step, we determine if there is a relationship between the inflexibility of the correlation matrix of a DNN's activations, which we consider a proxy of its covariability structure, and its robustness to adversarial perturbations. To this end, we analyse the correlation structure between the neural activations of a DNN in response to data which is perturbed with perturbations of different sizes. First, we train two 5-layer MLP models on FMNIST, one on clean data and the other via adversarial training. Then, we compute the correlation between the activations of the penultimate layer in response to clean and adversarially perturbed images. We use $\mathbf{R}_\epsilon$ to refer to the correlation matrix produced by data perturbed by perturbations of ℓ_∞ norm ϵ. We quantify the overall change in the correlation structure as $\|\mathbf{R}_0 - \mathbf{R}_\epsilon\|_F$, where $\|\cdot\|_F$ is the Frobenius norm and plot this quantity for several values of ϵ in Figure 6.1a. The correlation structure of the adversarially trained MLP is much more invariant to adversarial perturbations compared to the correlation structure of the MLP trained on clean data. It is only after the size of the perturbation becomes very large does the correlation structure of the adversarially trained model begins to change significantly. To verify that the change in the norm is not caused due to a small number of neurons, we compute the absolute change in the correlation of each neuron pair due to the addition of adversarial perturbations of size 0.1, and plot the cumulative frequency curve shown in Figure 6.1b. We see that the curve for the adversarially trained model is significantly shifted to the left of the curve for the model trained on clean data indicating that the correlation between most, if not all, pairs of neurons has not changed significantly.

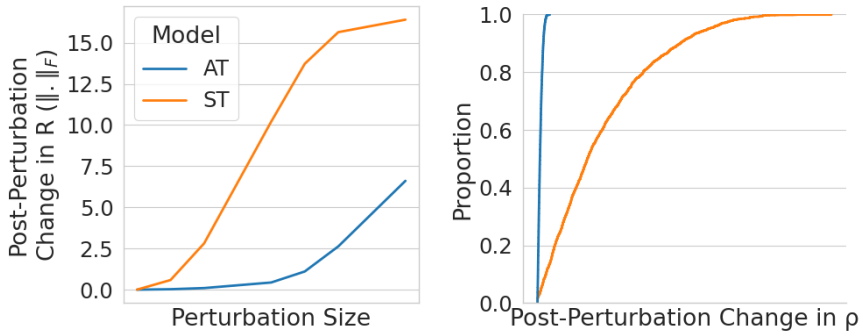
From these observations we can infer that the invariance of the inter-neuron covariability structure, across different perturbations of the input, is related to the adversarial robustness of the model. If this relationship is causal, then constraining the inter-neuron covariability structure should induce adversarial robustness. To validate this, we design a neural network layer that explicitly optimizes its activations to make them conform to a fixed covariability structure. We then include this layer in a DNN model and evaluate its robustness against state-of-the-art adversarial attacks.

6.3 Self-Consistent Activation Layer

We have developed the Self-Consistent Activation (SCA) layer to simulate an inflexible inter-neuron covariability structure. At a high-level, the SCA layer computes its output as $\text{SCA}(\mathbf{x}) = g_{\mathcal{C}}(\mathbf{a}_\mathbf{x})$ where $\mathbf{x} \in \mathbb{R}^{d_\mathbf{x}}$ is the input, $\mathbf{a}_\mathbf{x} = f(\mathbf{x}) \in \mathbb{R}^{d_\mathbf{a}}$ is the feed-forward activation vector, and $g_{\mathcal{C}}(\mathbf{a}_\mathbf{x})$ is the projection of $\mathbf{a}_\mathbf{x}$ onto $\mathcal{C}$, the subspace comprised of the vectors that respect the learned covariance structure. If we consider covariability to be a linear relationship, like covariance, then $g_{\mathcal{C}}$ would simply be a linear projection. However, to allow for more complex inter-neuron interaction, in this paper we have decided to adopt the following non-linear form for $g_{\mathcal{C}}$:

$$\arg\min_{\mathbf{a}_\mathbf{x}} \|\mathbf{a}_\mathbf{x} - \phi(\mathbf{W}_g\mathbf{a}_\mathbf{x} + \mathbf{b}_g)\|_2^2 + \lambda \|\mathbf{x} - \mathbf{W}_h\mathbf{a}_\mathbf{x} - \mathbf{b}_h\|_2^2, \tag{6.1}$$

where $\phi = ReLU$, $\mathbf{W}_g \in \mathbb{R}^{d_\mathbf{a} \times d_\mathbf{a}}$, $\mathbf{W}_h \in \mathbb{R}^{d_\mathbf{x} \times d_\mathbf{a}}$, $\mathbf{b}_g \in \mathbb{R}^{d_\mathbf{a}}$ and $\mathbf{b}_h \in \mathbb{R}^{d_\mathbf{x}}$. The first term represents the distance between the activation and its projection onto $\mathcal{C}$, while the second term



(a) (b)

Figure 6.1: (a) The Frobenius norm of the change in the correlation matrix of the activations of neurons in the penultimate layer of a MLP trained on clean (ST) and adversarially perturbed (AT) FMNIST images when adversarial perturbations of different sizes are added to the input; (b) CDF of the change in correlation between neuron pairs when adversarial perturbation of ℓ_∞ norm 0.1 is added.

represents the information about $\mathbf{x}$ that is not carried by $\mathbf{a}_\mathbf{x}$. The latter is added as a regularizer to prevent degenerate solutions, like $\mathbf{a}_\mathbf{x} = 0$, in which $\mathbf{a}_\mathbf{x}$ carries no information about $\mathbf{x}$, and λ is a scalar that controls the strength of the regularization. We set the diagonal of $\mathbf{W}_g$ to zero to prevent it from becoming the identity matrix and we perform the minimization using batch gradient descent. The exact sequence of operations performed by the SCA layer is shown in Algorithm 1.

Algorithm 1: SCA Layer

$\mathbf{u} \leftarrow f(\mathbf{x})$
for $t : 1 \rightarrow T$ **do**
    $\mathbf{a}_\mathbf{x} \leftarrow \phi(\mathbf{u})$
    $J \leftarrow \|\mathbf{a}_\mathbf{x} - \phi(\mathbf{W}_g\mathbf{a}_\mathbf{x} + \mathbf{b}_g)\|_2^2 + \lambda \|\mathbf{x} - \mathbf{W}_h\mathbf{a}_\mathbf{x} - \mathbf{b}_h\|_2^2$
    $\mathbf{u} \leftarrow \mathbf{a}_\mathbf{x} - \eta\nabla_{\mathbf{a}_\mathbf{x}} J$
end for
$\mathbf{a}_\mathbf{x} \leftarrow \phi(\mathbf{u})$

6.4 Evaluation

6.4.1 Experimental Setup

Datasets: We evaluate the performance of SCA layers on image and audio classification tasks. For image classification we use the MNIST LeCun et al. [2010] and Fashion MNIST (FMNIST) Xiao et al. [2017] datasets, which contain 60K 28×28 black-and-white images of handwritten digits and 10 types of apparel, respectively. From both MNIST and FMNIST, we use 45K images for training, 5K for evaluation, and 10K for testing. For the audio classification task we the SpeechCommands dataset Warden [2018], which contains around 100K 1 second, 16KHz recordings of humans vocalizing 35 commands. We use 84K recordings for training, 10K recordings for validation, and 9.6K recordings for testing.

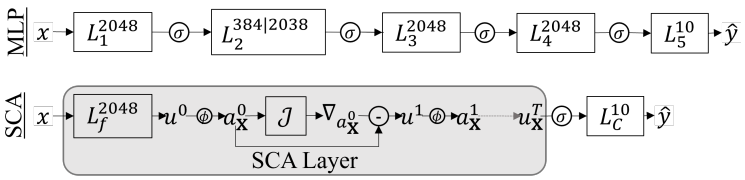
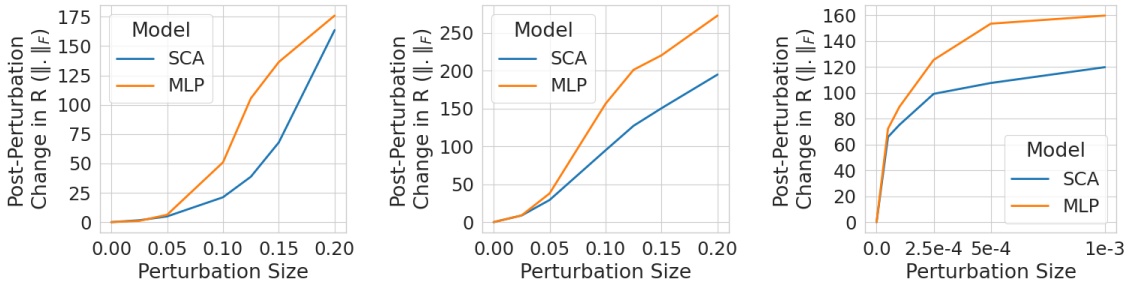


Figure 6.2: Schematics of the MLP and SCA models. $L_.$s are affine projections with the superscripts representing the output dimension. The output dimension of L_2 is set to 384 in MLPs trained on MNIST and FMNIST, and to 2048 in MLPs trained on SpeechCommands. $\phi = ReLU$, $\sigma =$ dropout $\circ\, \phi$, $\mathcal{J}$ is the loss from eq. (6.1).



(a) MNIST (b) FMNIST (c) SpeechCommands

Figure 6.3: The Frobenius norm of the change in the correlation matrix of the penultimate layer activations from the SCA model and MLP due to the addition of adversarial perturbations of different ℓ_∞ norms.

Data Preprocessing: For the image datasets, we flatten the image into a vector, which is then normalized by subtracting 0.5 and then dividing by 0.5. The audio data is preprocessed by first downsampling to 8KHz. Then 128 Mel-Frequency Cepstral Coefficients (MFCCs) are computed from a log mel spectrogram having 512 FFT points computed over a 64 ms sliding window with a stride of 32ms. By retaining only the first 16 MFCCs we obtain a 16×251 matrix for each 1s audio recording. The matrix is then flattened, and normalized by subtracting -0.96 and then dividing by 9.2 (the mean and standard deviation computed over the validation set).
Models: We compare the performance of models containing SCA layers (SCA model) to MLP models having comparable architectures and number of parameters. The schematics of these models are shown in Figure 6.2. The SCA model performs $T = 16$ optimization steps. The probability of dropout is set to 0.2 for all models. The models are optimized using the Adam optimizer using a learning rate of 0.001 and a batch size of 256 for up to 100 epochs. The learning rate is halved if the loss on the validation set does not decrease for 5 epochs, and if it does not decrease for 20 epochs the training is stopped early. All the results presented below are averaged over 5 trials with different random seeds.

6.4.2 Results

Analysis of Activation Covariability Structure To verify that SCA layers increase the invariance of the inter-neuron correlation structure, we analyse the correlation between the activations of the

Dataset	Model	Clean	Non-Adv Perturb	Adv Perturb Sizes (ℓ_∞) 0.05	0.1	0.125	0.15
MNIST	MLP	**98.4**	83.6	**90.0**	52.6	28.2	12.8
	SCA	97.9	**85.1**	88.1	**54.8**	**32.3**	**15.7**
FMNIST	MLP	88.7	75.9	39.2	7.7	3.1	1.0
	SCA	**89.4**	**76.4**	**46.9**	**12.7**	**6.1**	**2.8**
				Adv Perturb Sizes (ℓ_∞) 5e-5	1e-4	2.5e-4	5e-4
Speech	MLP	**81.4**	**52.0**	23.2	9.1	1.1	0.2
Commands	SCA	80.1	47.8	**27.9**	**15.3**	**3.0**	**0.5**

Table 6.1: The accuracy achieved by the SCA models and the baseline MLPs under adversarial and non-adversarial perturbations.

penultimate layer using the method introduced in 6.2. Specifically, we compute the correlation matrix $\mathbf{R}_\epsilon$ from the activations of the penultimate layer of the SCA model and MLP in response to 1000 data samples. These samples are perturbed by adversarial perturbations of ℓ_∞ norm ϵ. We compute $\mathbf{R}_\epsilon$ for each dataset using several values of ϵ. For each dataset and ϵ we then compute $\|\mathbf{R}_0 - \mathbf{R}_\epsilon\|_F$ to represent the overall change in the correlation structure due to the addition of adversarial perturbation of size ϵ. Figure 6.3 shows this quantity for the SCA model and MLP on each dataset. In every case the correlation structure of the SCA model changes more slowly than the MLP, and thus is more invariant to adversarial perturbation. This result shows that the SCA layer indeed produces the intended effect of constraining the covariability structure of neural activations.

Robustness of Models Trained on Clean Data

We evaluate the robustness of the SCA models by training them on *clean* data and computing their classification accuracy on adversarially and non-adversarially perturbed data. We compute adversarial perturbations of various ℓ_∞ norms using AutoAttack Croce and Hein [2020c], an ensemble of white- and black-box adversarial attacks. For non-adversarial perturbations, we use a set of common image and audio transforms at 4 levels of severity. The set of image perturbations includes Gaussian ($\sigma \in \{2^{-3}, 2^0\}$) and Uniform ($\{[0, \frac{b}{10}] | b \in [1, 4]\}$) noise, Gaussian blur ($\sigma \in [1, 4]$), rotation ($[2^1, 2^4]$ deg) and random occlusion ($\{12.5\%, 25\%, 50\%, 75\%\}$). The set of audio perturbations includes Gaussian, Uniform, and environmental noise (SNRs$\in [2^1, 2^4]$ dB), Room Impulse Response (RIR), speed manipulation ($\times\{1.75, 1.5, 1.25, 0.75\}$) and pitch shift (steps$\in [1, 4]$). We use the isotropic RIR from Ko et al. [2017] and environmental noise from Reddy et al. [2019].

Table 6.1 shows that SCA model is *significantly* more robust than the MLP. Most notably we see that the SCA model is more robust than the MLP to state-of-the-art adversarial attacks of various strengths. On average, the SCA model improves accuracy by 4.4%, 3.2%, and 1.8% absolute (93%, 105%, and 10% relative), compared to the MLP model on FMNIST, SpeechCommands and MNIST, respectively. Moreover, the SCA layer also makes the image classification models more robust to non-adversarial perturbations. While we do not show the breakdown here, the SCA models achieve higher accuracy on all types of non-adversarial image perturbations. However, the accuracy of the SCA model is lower than the MLP against non-adversarial audio perturbations. Further investigating this is part of future work. The above results clearly show that SCA layers

are much more robust to adversarial attacks than layers of perceptrons, and that their robustness is not limited to a particular type of data but generalizes across data complexity and modality.

Chapter 7

Adding Lateral and Top-Down Recurrence

7.1 Problem and Motivation

Another cognitive mechanism that we study is recurrent connectivity in the biological brain. Most modern DNNs, particularly those that are commonly employed for computer vision applications, process the input in a feed-forward manner – each neuron in a layer receives inputs only from neurons in the previous layer(s). On the other hand, in the primate visual system, the neurons are connected in a highly recurrent manner – neurons may receive inputs from neurons either the same, any preceding or any succeeding visual area [Bullier et al., 2001, Briggs, 2020]. This recurrent processing has been linked to the ability of primates to perform accurate object recognition under distortions such as crowding and occlusions [Spoerer et al., 2017]. While Kubilius et al. [2018] have proposed to simulate biologically plausible recurrence in DNNs, their work is limited to feeding the output of a DNN layer or module back into itself. We extend this body of work by integrating recurrent circuits between neurons in the same layer (lateral recurrence), as well as between neurons from different layers (feed-back recurrence).

7.2 Introducing Recurrence in CNNs

7.2.1 Overview

We introduce lateral and feedback recurrence into strictly feed-forward Convolutional Neural Networks (CNNs). The lateral connections feed the output of an intermediate convolution layer back to itself, while the feed-back connections feed the outputs of later layers to the earlier ones. The architecture of the resulting DNN is illustrated in Figure 7.1. The processing performed by the feed-forward pathways of the original CNN remains unchanged, however, due to the addition of lateral and feed-back pathways, the inputs to the intermediate convolution layers are modified by the outputs of succeeding layers as well as its own from the past. To perform the recurrent computations, we unroll the loops introduced by the recurrent connections for a fixed number of time steps, as shown in 7.1B. At each time step the model receives the original input image. We provide details of the processing performed by our model below.

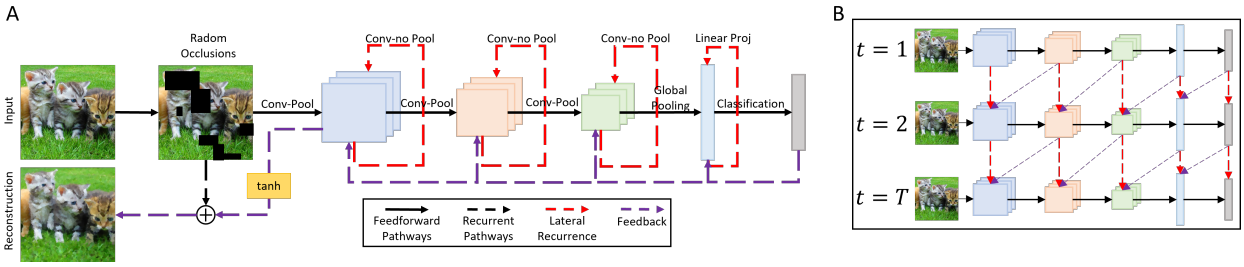


Figure 7.1: The DNN architecture after adding lateral and feed-back recurrence to a CNN (panel A). To perform the recurrent computations we unroll the loops present in the architecture for a fixed number of time-steps. This results in the computation graph illustrated in panel B.

7.2.2 Architectural Details

A conventional feed-forward CNN is a cascade of convolution (cross-correlation) and pooling operations, interleaved by non-linear activations, followed by a linear classifier. Formally, this can be written as

$$F_f(x) = C(z^{(L)}) \quad \text{s.t } z^{(l)} = \sigma\left(\phi^{(l)}(z^{(l-1)})\right),\ z^{(0)} = x, \tag{7.1}$$

where x is the input image, σ is the non-linear activation function, and $\phi^{(l)}$ represent convolutional (and pooling) layer l.

In contrast, our proposed recurrent CNN architecture, after being unrolled through time, performs the following computation

$$F_r(x) = C(z^{(L,T)}) \quad \text{s.t } z^{(l,t)} = \sigma\left(\phi^{(l)}(\sigma\left(z^{(l-1,t)} + \tilde{z}^{(l,t-1)} + \tilde{z}^{(l+1,t-1)}\right)\right),\ z^{(0,t)} = x,\ z^{(l,0)} = 0, \tag{7.2}$$

where $z^{(l,t)}$ represents the feature map produced by layer l at time t. Here $z^{(l-1,t)}$ represents the feed-forward signal arising from the output of the preceding layer, $\tilde{z}^{(l,t-1)}$ represents the lateral signal arising from the output of layer l at the previous time step, and $\tilde{z}^{(l+1,t-1)}$ represent the feedback signal, arising from the succeeding layer at the previous time step. The lateral and feedback signals are computed as follows:

$$\tilde{z}^{(l,t-1)} = \psi^{(l)}(z^{(l,t-1)}) \qquad \tilde{z}^{(l+1,t-1)} = \psi^{(l+1)}(z^{(l+1,t-1)}), \tag{7.3}$$

where ψ represent the transposed convolution layers that linearly project and upsample the feature maps such that the spatial dimensions of $z^{(l-1,t)}$, $\tilde{z}^{(l,t-1)}$ and $\tilde{z}^{(l+1,t-1)}$ match. Like the conventional CNN, our proposed model can be trained using backpropagation with standard loss functions.

Chapter 8

Biologically Inspired Speech Recognition

8.1 Problem and Motivation

Given that we achieved encouraging results from integrating visual sensory and cognitive biological priors in DNNs, we propose to extend this approach to the auditory domain as well. Specifically, we seek to integrate biologically plausible feature extraction and processing within Automatic Speech Recognition (ASR) DNNs. It has been found, albeit in relatively simple settings, that using more biologically plausible features results in more robust ASR Stern and Morgan [2012]. However, to the best of our knowledge, the robustness, especially to adversarial attacks, of such features has not been studied in conjunction with modern ASR models trained on large and diverse speech data. To fill this gap, we study the impact on transcription accuracy and robustness of using acoustic features that are more biologically plausible than those commonly used for ASR (i.e. Log Mel Spectrogram).

8.2 Biologically Plausible Speech Features

In this Section, we describe the acoustic features that we consider in this study. An overview of the computations involved in these features is presented in Figure 8.1. We only consider spectral features and thus the initial processing for all the features involves computing the audio signal's Short-Time Fourier Transform (STFT). Thereafter, the processing for each feature varies as explained below.

8.2.1 Features From Prior Works

Log Spectrogram (LogSpec) is obtained by applying the log nonlinearity to the STFT.

Log Mel-Spectrogram (LogMelSpec) is obtained by applying the Mel filterbank to the STFT followed by the log nonlinearity.

Mel Frequency Cepstral Coefficients (MFCC) are computed by applying the Discrete Cosine Transform (DCT) to the LogMelSpec.

Gammatone Spectrogram (GammSpec) is obtained by applying the *normalized* gammatone filterbank to the STFT followed by the cube root nonlinearity. Each filter is normalized such that its coefficients sum to 1. Fig. 8.2 shows a sample GammSpec.

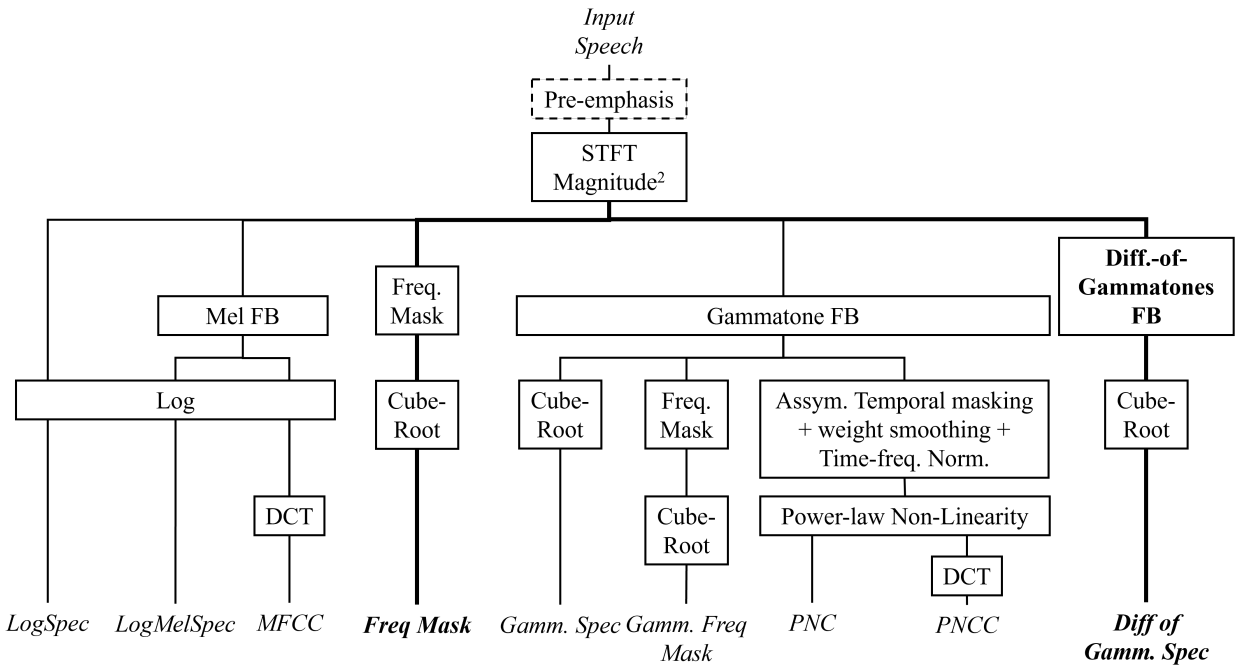


Figure 8.1: Overview of acoustic feature computation. Novel features in bold.

Power Normalized Coefficients (PNC) Kim and Stern [2016] combine temporal masking, weight smoothing, and time-frequency normalization to achieve more robust ASR, and are computed as follows. First, the audio signal is pre-emphasized (coeff.=0.97), and the magnitude-squared spectrum is computed via STFT. Then a squared normalized gammatone filterbank, i.e. the filterbank coefficients are squared, and divided by the sum (after squaring), is applied to the spectrum to obtain the short-time spectral power, $P[m, l]$, where m and l represent frame and channel indices. The medium-time power is also computed as $Q[m, l] = \frac{1}{2M+1} \sum_{i=-M}^{M} P[m + i, l]$. Next, asymmetric noise suppression is applied to Q to obtain Q_{le}. The difference between Q and Q_{le} is computed and half-wave linear rectified. This is followed by temporal masking, which is the phenomenon that causes an audio signal to become imperceptible if it is temporally adjacent to a louder signal. To simulate temporal masking, first the online peak power, $Q_p[m, l]$, is computed as a running maximum over Q_0. Then the masked signal, $R[m, l] = Q_0[m, l]$ if $Q_0[m, l] \geq \lambda_t Q_p[m - 1, l]$ else $\mu_t Q_p[m - 1, l]$, where $\lambda_t = 0.85$ and $\mu_t = 2$ in Kim and Stern [2016]. Next, spectral weight smoothing and mean power normalization are applied. Finally, a rate-level non-linearity (raise to power $1/15$) is applied.

Power Normalized Cepstral Coefficients (PNC) Kim and Stern [2016] are computed by apply the DCT to PNC features.

8.2.2 Novel Features

Frequency Masked Spectrogram (FreqMask) simulates simultaneous frequency masking, which is the phenomenon that a loud frequency (the *masker*) can *mask* adjacent quieter frequencies rendering them inaudible Lin et al. [2015].

We start with a STFT spectrogram, compute the masking threshold for all frequencies at each

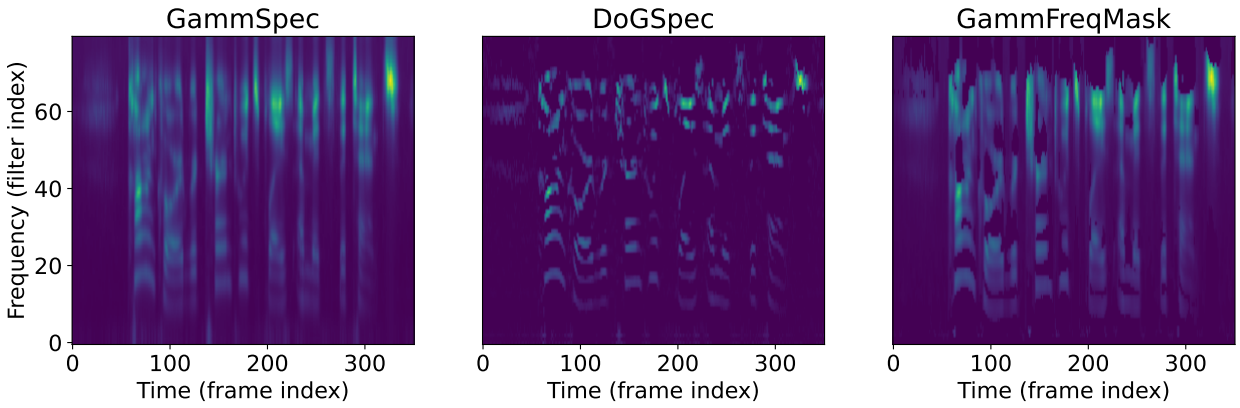


Figure 8.2: Gammatone (GammSpec), Difference-of-Gammatone (DoGSpec) and Frequency Masked Gammatone (GammFreqMask) spectrograms.

time step, and zero the energy in frequencies with level below the masking threshold. We use the method from Lin et al. [2015], Qin et al. [2019] to estimate the masking threshold as follows. First, the log-magnitude power spectral density (PSD) is computed for each STFT frame, x, and frequency bin, k, as $p_x(k) = 10\log_{10}\left|\frac{s_x(k)}{N}\right|^2$, where $s_x(k)$ is the spectral magnitude in STFT bin k of frame x, and N is the window size (in samples) used to compute the STFT.

The PSD is then normalized to have a maximum sound pressure level (SPL) of 96 dB, this is referred to as the normalized PSD, $\bar{p}_x(k) = 96 - \max_k\{p_x(k)\} + p_x(k)$. Next, the normalized PSD is smoothed by its neighbors: $\bar{p}_x^m(k) = 10\log_{10}\left[10^{\frac{\bar{p}_x(k-1)}{10}} + 10^{\frac{\bar{p}_x(k)}{10}} + 10^{\frac{\bar{p}_x(k+1)}{10}}\right]$

We then compute the masking threshold induced by the masker frequency f_i on frequency f_j as: $T[b(i), b(j)] = \bar{p}_x^m(b(i)) + \Delta_m[b(i)] + \text{SF}[b(i), b(j)]$, where

- $b(i)$ is the bark scale of frequency f_i[1],
- $\Delta_m[b(i)] = -6.025 - 0.275b(i)$,
- $G(b(i)) = -27 + 0.37\max\{\bar{p}_x^m(b(i)) - 40, 0\}$,
- $\Delta b_{ij} = b(i) - b(j)$ and
- $\text{SF}[b(i), b(j)] = 27\Delta b_{ij}$ if $\Delta b_{ij} > 0$ else $G(b(i)) \cdot \Delta b_{ij}$

Finally, for each frequency, f_j, we can compute the global masking threshold by combining $T[b(i), b(j)]$ for all i as

$$\theta_x(j) = 10\log_1 0\left[10^{ATH(j)/10} + \sum_i 10^{T[b(i),b(j)]/10}\right], \tag{8.1}$$

where ATH[2] is the minimum PSD a frequency must have to be perceptible *in quiet*. Frequency masking is applied to the spectrogram by setting the spectral magnitude to zero at frame x and frequency bin j if $\bar{p}_x^m(j) < \theta_x(j)$. Finally, Cube root nonlinearity is applied.

[1] frequency to bark: $b(f) = 13\arctan(0.00076f) + 3.5\arctan(f/7500)$.

[2] $ATH(f) = 3.64\left(10^{-3}f\right)^{-0.8} - 6.5e^{-0.6(10^{-3}f-3.3)^2} + 10^{-15}f^4$

Frequency Masked Gammatone Spectrogram (GammFreqMask) applies FreqMask to GammSpec (before the nonlinearity). Fig. 8.2 shows a sample GammFreqMask.

Difference of Gammatone Spectrogram (DoGSpec) simulates lateral suppression, the phenomenon that the response to a frequency may be suppressed if adjacent frequencies are present in the signal Stern and Morgan [2012], even if the intensity of the latter is below the threshold of hearing. While lateral suppression has been simulated in several cochlear models in prior work Lyon [1984], Slaney [1988], these models are not amenable to be used in modern DNN-based systems because they tend to be computationally intensive and rather slow. As a result, lateral suppression is largely missing from modern ASR systems. To fill this gap we have developed the DoGSpec feature for incorporating lateral suppression into DNNs.

The key component in DoGSpec is the DoG filterbank, which is constructed as follows. First, two normalized gammatone filterbanks, G_1 and G_α, are created such that the bandwidths of the filters in G_1 are scaled α to obtain the filters in G_α. Next, the corresponding filters in the two filterbanks are subtracted to obtain $G_d[i,j] = G_1[i,j] - G_\alpha[i,j]$. The filters are normalized by dividing the coefficients by the sum of the positive coefficients to ensure the excitatory components sum to 1, i.e. $\bar{G}_d[i,j] = \frac{G_d[i,j]}{\sum_{j'=0}^{F-1} \max(G_d[i,j'],0)}$. G_1, G_α and $\bar{G}_d$ are shown in Figure 8.3. Note that the DoG filter has negative coefficients on the frequencies adjacent to the center frequency and, thus, any energy in those frequencies will suppress the response of the filter. Given an input audio signal, the DoGSpec is computed by applying pre-emphasis, then the DoG filterbank and finally a cube-root non-linearity. Fig. 8.2 shows a sample DoGSpec.

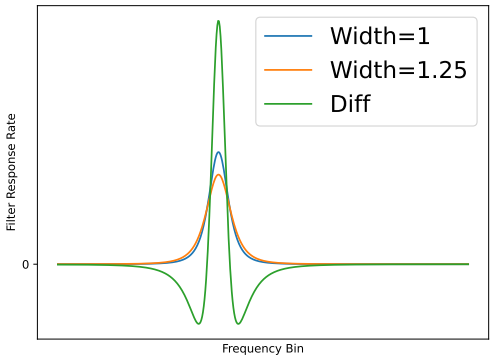


Figure 8.3: DoG Filter

8.3 Evaluation Setup

To evaluate the robustness and accuracy of the features described above, we train various ASR models with these features on diverse datasets and then evaluate them on clean and noisy data. The details of the evaluation setup are as follows.

8.3.1 Train Datasets

We train models on three datasets, namely LibriSpeech Panayotov et al. [2015], TEDLIUM Hernandez et al. [2018], and the Spanish subset of Multilingual LibriSpeech (MLS-es) Pratap et al. [2020]. LibriSpeech and MLS-es contains read speech from audio books in English and Spanish, respectively, while TEDLIUM contains spontaneous English speech from recorded TED talks. We use the full 960 hour and 452 hour training sets of LibriSpeech and TEDLIUM, respectively, for training. For MLS-es, we use the 917 hour train set from Huggingface.

8.3.2 Models

We use recipes from SpeechBrain Ravanelli et al. [2021] to train 13M parameter Conformer Gulati et al. [2020] and 104M parameter Branchformer Peng et al. [2022] ASR models. The LibriSpeech and MLS-es models use a 5k subword unigram tokenizers trained, while the TEDLIUM models use a 500 BPE tokenizer. The models trained on LibriSpeech also contain a transformer LM for re-scoring during decoding. During robustness evaluation, all models use beam size 10 without LM rescoring.

The original recipes use LogMelSpec features so we minimally modify the recipes to incorporate the various features from §**??**. Since we set the number of filters to 80 for all filterbank-based features, we only change the `compute_feature` field in the recipe YAML while the downstream model remains unchanged. For LogSpec and FreqMask however, we do need to change the input size of the transformer because these features do not use filterbanks.

8.3.3 Evaluation Setup

Methodology and Data

We evaluate the models' accuracy on the training dataset's official test subsets. To evaluate robustness we use Speech Robust Bench (SRB) Shah et al. [2025], a recently released robustness benchmark for ASR models. SRB contains multi-lingual speech with more than 100 types of noises and distortions. At a high-level, the distortions in SRB fall into 5 categories: inter-personal communication (drawn from CHiME Barker et al. [2017] and AMI Kraaij et al. [2005] corpora), environmental effects (environmental noise from ESC-50, MS-SNSD, MUSAN and WHAM) and room impulse responses from Kinoshita et al. [2013]), digital augmentations (white noise, special effects, audio processing operations like resampling, gain and filtering), speech variations (accented speech from CommonVoice Ardila et al. [2019], and text-to-speech using YourTTS Casanova et al. [2021]), and adversarial attack (SNR-bounded untargeted PGD Madry et al. [2018a]). The PGD attack perturbs the audio to maximize the transcription loss (CTC or NLL) while keeping the SNR above a specified lower bound. We also evaluate the models on the targeted "imperceptible" attack Qin et al. [2019], which unlike PGD, generates the minimal (but unbounded) perturbation that causes the model to output a specified target string response to the input audio. This attack exploits frequency masking to add noise in the spectral regions that are likely to be masked by human hearing, and, thus, is a good test for FreqMask and DoGSpec. Since this attack is very computationally expensive, we evaluate only LibriSpeech models against it using 10 utterances from LibriSpeech test-clean.

Metrics

To measure accuracy we use *Word Error Rate* (WER). To measure robustness we use *WER Degradation* (WERD) and *Normalized WERD* (NWERD) Shah et al. [2025]. WERD is computed as the difference between the model's WER on the clean test subset of its training dataset and its WER on the noisy testing data. NWERD is computed by dividing WERD by a measure of speech quality, specifically DNSMOS Reddy et al. [2020] and/or PESQ Rix et al. [2001], such that errors on less distorted utterances are penalized more than errors on more distorted utterances. This is done because a robust model should not compromise accuracy on cleaner utterances, which represent the

Dataset/ Model	Feature	Test-clean (bs=1)	Test-clean (wLM, bs=66)	Test-other (wLM, bs=66)
LibriSpeech/ Conformer	LogMelSpec	5.25	2.53	6.04
	DoGSpec	5.27	2.50	6.17
	FreqMask	5.71	2.72	7.85
	GammFreqMask	5.02	2.46	6.32
	GammSpec	**4.65**	**2.29**	**5.62**
	LogSpec	4.66	2.30	5.82
	MFCC	8.31	3.23	9.44
	PNCC	8.35	3.42	9.44
	PNC	5.69	2.50	6.70
LibriSpeech/ Branchformer	LogMelSpec	3.66	**2.01**	**4.78**
	DoGSpec	3.12	2.11	5.21
	GammSpec	**3.11**	2.07	5.25
		Test (bs=1)	Test (bs=66)	
TEDLIUM/ Branchformer	LogMelSpec	15.07	7.52	
	DoGSpec	16.70	8.21	
	GammSpec	**15.86**	**8.06**	
MLS-es/ Branchformer	LogMelSpec	6.20	6.11	
	DoGSpec	6.43	6.19	

Table 8.1: The WER of ASR models with different features on the original test sets of LibriSpeech, TEDLIUM and MLS-es. bs is the beam size and wLM indicates rescoring with an LM

average use case while improving accuracy on severely distorted utterances. To evaluate robustness against the targeted imperceptible adversarial attack Qin et al. [2019], we compute the WER between the target phrase and the predicted transcript. Since the attack is unbounded, we also consider the *Signal-to-Noise Ratio* (SNR) of the perturbation because it can potentially transform the input audio into an utterance of the target string. We consider a model to be robust to this attack if either the WER is high or the SNR is low.

8.4 Results

8.4.1 Accuracy on Clean Utterances

Table 8.1 shows the WER of the various models and features on the unmodified test subsets of LibriSpeech, TEDLIUM and MLS-es. We observe that despite being default features of choice in modern ASR systems including models like Whisper Radford et al. [2023] and Canary Elena Rastorgueva LogMelSpec is outperformed by GammSpec on all datasets. Interestingly, LogSpec performs similar to GammSpec even though it has 5 times the dimensionality, which indicates that GammSpec retains most of the relevant information present in the raw spectrogram. The effectiveness of the gammatone filterbank is further evidenced by the fact that GammFreqMask has

Model	Feature	SNR	WER
Branchformer	DoGSpec	13.60	11.03
	LogMelSpec	25.10	5.15
Conformer	DoGSpec	**8.60**	0.00
	FreqMask	10.80	**19.12**
	GammFreqMask	10.10	7.35
	GammSpec	15.20	13.24
	LogMelSpec	21.30	6.62
	LogSpec	15.00	3.68
	PNC	13.40	8.09

Table 8.2: The WER between the prediction and target phrase for the various models and features and the SNR of the adversarially perturbed audio.

much lower WER than FreqMask. Furthermore, while we expected FreqMask, GammFreqMask and DoGSpec to degrade WER because they discard some spectral information, we note that the degradation is minimal if any. In fact, on LibriSpeech test-clean both DoGSpec and GammFreqMask outperform LogMelSpec under beam-search decoding. On TEDLIUM and MLS-es, however, DoGSpec has slightly higher WER than LogMelSpec. As we shall see in the following sections, what DoGSpec lacks in accuracy, it makes up for in robustness.

8.4.2 Robustness to Adversarial Attacks

We consider two adversarial attacks in our evaluation: the untargetted PGD attack and targeted "imperceptible" attack. The PGD attack perturbs the audio to maximize the transcription loss (CTC or NLL) while keeping the SNR above a specified level. We use WERD. The targeted attack generates the minimal perturbation that causes the model to output a specified target string response to the input audio.

Figure 8.4 shows the WERD of the features and models against the PGD attack under different SNR bounds. We exclude PNCC and MFCC from this analysis because they achieved very high WERs on clean data. We observe that DoGSpec achieves the lowest WERD across all SNR bounds, followed by LogSpec. Meanwhile, LogMelSpec performs the worst. Interestingly, FreqMask, GammFreqMask, PNC and GammSpec perform similarly, and have much greater WERD than DoGSpec and LogSpec. Turning to Table 8.2, we see that attacks on DoGSpec have the lowest SNR. Under these SNR values, the "imperceptible" attack is going to be very perceptible and will likely significantly impact speech intelligibility, thus violating the basic requirements of an effective adversarial attack. GammFreqMask and FreqMask also have fairly low SNR values, and FreqMask also has high WER. These results show that our proposed features significantly improve robustness to targeted attacks. Meanwhile, LogMelSpec performs the worst a margin allowing the attack to achieve a low WER with 21.3 dB SNR, which may well be imperceptible, or barely perceptible. We are surprised to observe that PNC, despite being specifically designed to counteract noise, did not perform very well against adversarial attacks.

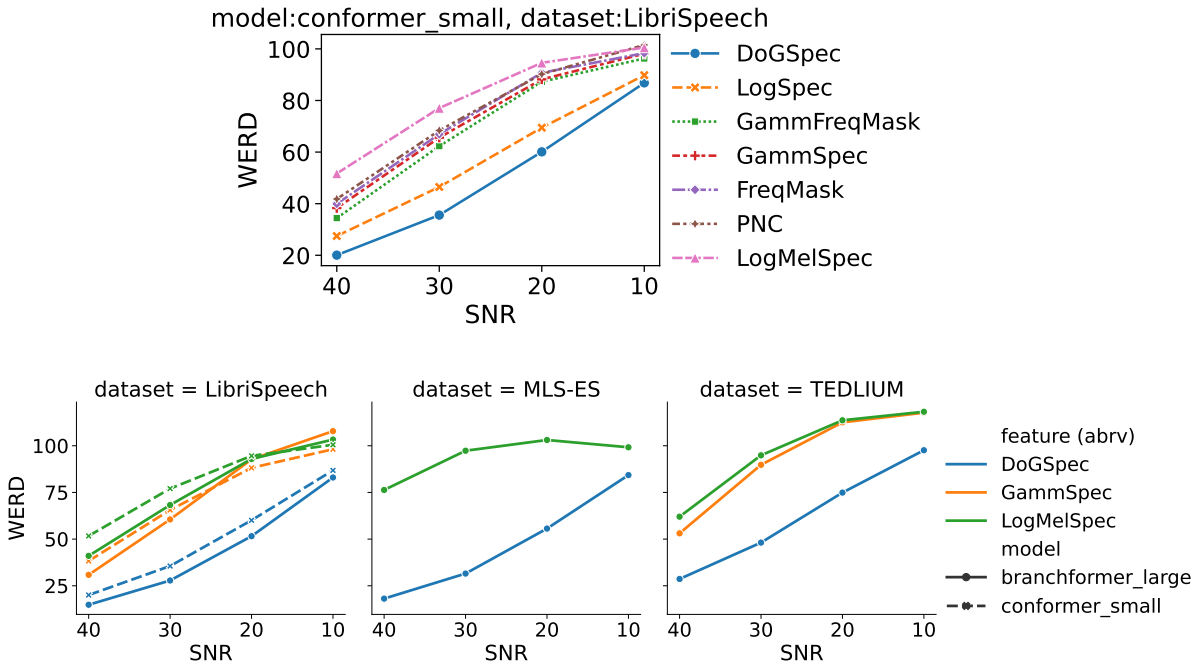


Figure 8.4: The WERD of models trained on various datasets and features against PGD attacks of increasing SNR bounds.

Dataset	Model	Feature	NWERD	WERD
LibriSpeech	Branchformer	DoGSpec	0.273	16.93
		GammSpec	**0.266**	17.37
		LogMelSpec	**0.266**	**16.54**
	Conformer	DoGSpec	0.35	21.03
		FreqMask	0.45	26.75
		GammFreqMask	0.36	22.21
		GammSpec	**0.33**	**20.02**
		LogMelSpec	0.37	23.73
		LogSpec	0.36	21.93
		MFCC	0.58	33.84
		PNC	0.38	22.84
		PNCC	0.54	30.99
TEDLIUM	Branchformer	DoGSpec	0.53	30.72
		GammSpec	**0.48**	29.42
		LogMelSpec	0.49	**28.89**
MLS-es	Branchformer	DoGSpec	0.71	28.44
		LogMelSpec	**0.59**	**25.59**

Table 8.3: WERD and NWERD on the SRB benchmark.

8.4.3 Robustness to Non-Adversarial Noise

We evaluate the models on non-adversarial noisy speech recordings from the SRB benchmark and present aggregated values for NWERD and WERD in Table 8.3. We see that for the conformer models trained on LibriSpeech and branchformer trained on TEDLIUM,GammSpec achieves the lowest NWERD, while the NWERD of DoGSpec is slightly higher than LogMelSpec. Interestingly, PNC has one of the highest NWERD which indicates that the noise reduction mechanisms included in it do not generalize to diverse types of corruptions.

Part III

Robustness Evaluation

Chapter 9

Robustness Benchmark For Speech Recognition Models

9.1 Problem and Motivation

As novel ML models continue to be developed and deployed at an ever-increasing rate, it has become crucial to ensure their robustness to challenging real-world scenarios, where corruptions arising from a myriad of sources, including the environment, sensing apparatus, and even malicious actors are present. To this end, prior works have developed comprehensive robustness benchmarks, particularly for vision [Hendrycks and Dietterich, 2019, Hendrycks et al., 2021a,b, Croce et al., 2020] and natural language processing models [Wang et al., 2021a, 2022b], that evaluate a model's performance under a variety of challenging scenarios. These benchmarks have proven to be invaluable to the advancement of research into more robust models because (1) they unify robustness evaluations, thus enabling meaningful comparisons across models and allowing progress to be accurately tracked, and (2) they make it easier for researchers to comprehensively evaluate the robustness of their models by aggregating a diverse and representative set of scenarios, and methods of simulating them, in a single benchmark.

While several robustness benchmark datasets exist for Automatic Speech Recognition (ASR) models [Barker et al., 2017, Kraaij et al., 2005, Wichern et al., 2019, Reddy et al., 2020, Cosentino et al., 2020, Hershey et al., 2016, Chen et al., 2020b, Snyder et al., 2015, Kinoshita et al., 2013, Ko et al., 2017, Nakamura et al., 2000, Jeub et al., 2009], none of the currently existing ones are in any sense comprehensive, because each benchmark measures the model's robustness w.r.t. to one or a few specific types of corruptions or scenarios, which puts the onus on model developers to find and collect all the relevant benchmarks to evaluate their model comprehensively. This has often resulted in model developers evaluating their models on disparate benchmarks [Radford et al., 2023, Wen et al., 2016, Chen et al., 2022, Likhomanenko et al., 2020], which makes it hard to reliably compare performance and robustness across models. Recently, Huggingface Open ASR Leaderboard [Srivastav et al., 2023] has sought to unify ASR model evaluations by developing a benchmark consisting of several real-world speech datasets. Although evaluating models on exclusively natural data may accurately reflect average case real-world performance, it is generally not informative about the specific types of corruptions the models are weak against, because the noise sources present in these datasets are not controlled or even fully known. For example, the

crowdsourced recordings in Common Voice [Ardila et al., 2019] contain a variety of distortions including sensor noise from low-quality equipment, background noise, and mispronunciation by non-native speakers. Furthermore, digital perturbations like special effects, computer-generated speech, and adversarial examples, that may be prevalent in digital content are largely overlooked by existing benchmarks.

We have developed `Speech Robust Bench` (`SRB`), a benchmark for comprehensively evaluating the robustness of ASR models to input perturbations and corruptions. `SRB` is designed to address the aforementioned major shortcomings of existing ASR robustness benchmarks, i.e., that (1) they are often specialized and thus are not individually comprehensive, (2) even taken together, they overlook important challenging scenarios, like special effects and adversarial attacks, and (3) may not reveal the specific weaknesses of the models. `SRB` addresses these shortcomings by evaluating ASR models under a comprehensive set of challenging scenarios, using recordings that are either are recorded under specific scenarios, and thus are inherently "noisy", or recordings that are digitally perturbed to simulate the various scenarios. `SRB` uses real recordings of accented speech and inter-personal conversations to evaluate robustness to articulatory and lexical variability. We take care to ensure that the recordings are clean and do not have any other corruption that may confound the results. To digitally simulate challenging scenarios, we curate a large comprehensive bank of 114 perturbations that represent common distortions arising from the environment, recording equipment, special effects, computer-generated speech, and adversarial attacks that are often overlooked by existing benchmarks.

To facilitate out-of-the-box robustness evaluations for the community, we have publicly released a large dataset [1] containing perturbed versions of LibriSpeech [Panayotov et al., 2015] test-clean, Spanish, and French and German test sets of Multilingual LibriSpeech [Pratap et al., 2020], as well as accented speech from common voice, and segemented near- and far-field audios from CHiME-6 [Reddy et al., 2020] and AMI[Kraaij et al., 2005]. We also release our code[2] with clear documentation to enable reproducibility and extensibility.

9.2 Robustness Benchmark For Speech Recognition Models

`Speech Robust Bench` (`SRB`) evaluates the robustness of ASR models by a three-step process consisting of (1) scenario simulation, (2) transcription, and (3) metrics computation, as shown in Fig. 9.1: first, various challenging speech recognition scenarios are simulated by applying a large bank of synthetic perturbations to clean speech datasets (§ 9.2.1), as well as by using inherently noisy speech datasets with limited and known sources of real noise and variations that are difficult to simulate.

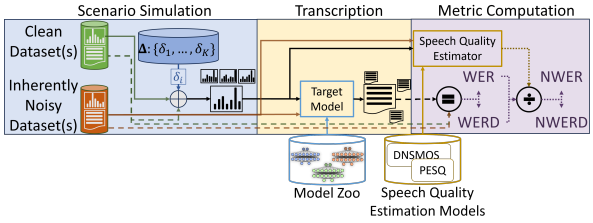


Figure 9.1: An illustration of the processes involved in using our benchmark to evaluate the robustness of ASR models.

Next, the perturbed recordings, the original clean recordings, and the recordings with inherent noise are transcribed using the target ASR model. Finally, the predicted and reference transcripts are compared, and the accuracy and

[1] data: https://huggingface.co/datasets/mshah1/speech_robust_bench_public

[2] code:https://github.com/ahmedshah1494/speech_robust_bench

robustness of each model in each setting is captured with various metrics (§ 9.2.2). To account for the differences in the level of difficulty between scenarios, we also estimate speech quality scores using appropriate models and use them to calculate normalized metrics.

9.2.1 Scenario Simulation

The various speech recognition scenarios simulated by SRB are taxonomized in Fig. 9.2, and can be divided into six high-level categories, namely (1) clean speech, (2) social gatherings, (3) speech variations, (4) environmental effects, (5) digital augmentations, and (6) adversarial attacks. The scenarios are described briefly below, while more details are given in Appendix 11.1.

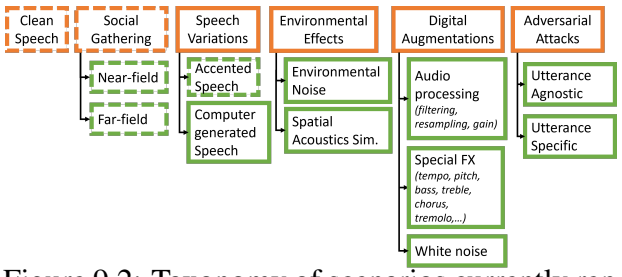


Figure 9.2: Taxonomy of scenarios currently represented in SRB. Scenarios in dashed boxes have real-world recordings, while scenarios in solid boxes are simulated by digitally adding perturbations.

(1) Clean speech: SRB uses clean speech for two purposes: to benchmark the baseline accuracy of ASR models, and to simulate various challenging scenarios by perturbing it. Clean speech is drawn from Librispeech [Panayotov et al., 2015] test-clean, TEDLIUM [Hernandez et al., 2018] release 3 test and MultiLingual LibriSpeech (MLS) [Pratap et al., 2020] test. LibriSpeech contains professional recordings of English audio books. Meanwhile, TEDLIUM contains professional recordings of English TED talks and provides lexical and phonetic diversity which LibriSpeech may lack. To increase the applicability of SRB to non-English and multi-lingual models we also include Spanish speech from MLS, which contains professionally recorded audio books in several languages.

(2) Social Gatherings: The ability to transcribe speech from semi-formal or informal settings, as well as far-field audio is useful for models used in meeting rooms, smart homes, and even subtitle generation, thus SRB includes English speech from dinner parties and meetings recorded by (2.1) *near-* and (2.2) *far-field* mics from CHiME-6 [Barker et al., 2017] and AMI [Kraaij et al., 2005].

(3) Speech Variations: ASR models must remain accurate under variations in pronunciation and prosody to serve diverse speakers. We therefore include (3.1) *clean accented speech*[3] from English and Spanish subsets of Common Voice 17 (CV17, Ardila et al. 2019) in SRB. To provide additional prosodic variability and to represent the increasing pervasiveness of generative AI, we also include synthetic speech generated by YourTTS [Casanova et al., 2022] (English) and Bark Suno [b] (Spanish) from transcripts of the three clean datasets from scenario (1) in the voices of English and Spanish speakers from VCTK [Yamagishi et al., 2019] and Bark Speaker Library (v2, Suno a), respectively.

The following scenarios involve synthetic perturbation of all three clean datasets from scenario (1).

(4) Environmental Effects: While noisy real speech datasets like CommonVoice [Ardila et al., 2019] and Switchboard [Godfrey et al., 1992] exist, the noise in them is not controlled or even known. Thus in SRB, we perturb clean speech to simulate (4.1) *environmental noise*, and (4.2)

[3]Accent annotations (excluding US English), and a DNSMOS score ≥ 3.4 [Reddy et al., 2020].

spatial acoustics. Concretely, we add *real* environmental noise from ESC-50 [Piczak, 2015], MS-SNSD [Reddy et al., 2019], MUSAN [Snyder et al., 2015] and WHAM! [Wichern et al., 2019] at Signal-to-Noise Ratios (SNR) of 10, 20, 30 and 40 dB. To simulate spatial acoustics, we add echo via SoX[4] and simulate Room Impulse Response (RIR) via convolution with real and simulated RIRs from Ko et al. [2017].

(5) Digital Augmentations: Digital media often undergoes processing and contains special effects, which are therefore included in SRB. Specifically, we include *standard audio processing operations* like amplitude gain, resampling, lowpass, and highpass filtering, (2.2) *special effects* like bass gain, treble gain, tempo increase, tempo decrease, speed increase, speed decrease, pitch increase, pitch decrease, chorus, tremolo, and phaser, and (2.3) *Gaussian white noise*.

(6) Adversarial Attacks: Models used in high-stakes settings are prime targets for adversaries and thus must resist attempts to compromise their accuracy. We use two types of adversarial attacks in SRB: (2.1) *utterance-specific* and (2.2) *utterance-agnostic* attacks. The utterance-specific attack searches for a perturbation δ, for a given speech recording x, such that a given model maximally mistranscribes it. To find δ, we follow Madry et al. [2018b] and use projected gradient descent to solve $\max_{\delta:\text{SNR}(\delta,x)\leq\epsilon} \mathcal{L}(M(x), y^*)$, where $\mathcal{L}$ is a differentiable loss function, like CTC-Loss, between the model's output $M(x)$ and the true transcript y^*, with $\epsilon \in [10, 40]$. The utterance-agnostic attack is similar to the utterance-specific attack, except δ is optimized over a held-out *set*, $\mathcal{X}^{\text{dev}}$, instead of each test utterance. This represents a more realistic scenario where an attacker tries to mount a denial-of-service attack against an ASR model by introducing utterance-agnostic perturbation at some point in the transcription pipeline. We use the method of Neekhara et al. [2019] to find $\delta : \mathbb{E}_{x\in\mathcal{X}^{\text{dev}}}\text{SNR}(\delta, x) \leq \epsilon \in [10, 40]$ such that $CER(\{x + \delta | x \in \mathcal{X}^{\text{dev}}\}) > \tau$ (see Alg. 2), where $\mathcal{X}^{\text{dev}}$ is the dev split of LibriSpeech, TEDLIUM and MLS.

Note of Extensibility and Usage: We have released our source code with instructions for reconstructing the data in SRB, and reproducing the results of this paper (§ **??**). SRB can easily be extended to other languages and speech datasets using the provided scripts for extracting accented speech from any language in CV17, and for simulating scenarios 3.2-6 on any speech recording or dataset.

We have also publicly released the data for all the above scenarios, except adversarial attacks and perturbed TEDLIUM recordings, on Huggingface Hub (see footnote 1). We made these exceptions because TEDLIUM's license prohibits the distribution of derivatives, and the adversarial attacks must be computed separately for each target ASR model. *To the extent possible, we encourage users to evaluate their models on the publicly released data to ensure reproducibility.*

9.2.2 Metrics

SRB measures the *utility* of the model with the widely used **Word Error Rate**. WER is computed as the word-level edit distance between the reference and the predicted transcripts, normalized by the length of the reference (see Appendix 11.2 for formal definitions). To measure the *robustness* of the model under challenging scenarios, we use **WER Degradation** (WERD), computed as $WER(\mathcal{X}_s) - WER(\mathcal{X})$, where $\mathcal{X}$ and $\mathcal{X}_s$ are datasets containing clean speech and speech from scenario s, respectively. For scenarios (1)-(3.1) (see § 9.2.1), $\mathcal{X}_s$ is an inherently noisy dataset,

[4]Available from https://sourceforge.net/projects/sox/.

and $\mathcal{X}$ will be LibriSpeech for English and Multi-Lingual LibriSpeech for Spanish. For scenarios (3.2)-(6), $\mathcal{X}$ is a clean dataset, and $\mathcal{X}_s$ is a perturbed version of $\mathcal{X}$.

When aggregating metrics (WER/WERD) over multiple scenarios, we follow the practice of Hendrycks and Dietterich [2019] and divide the metric by a measure of difficulty, i.e., by the (estimated) speech quality degradation. This adds weight to errors on "easy" scenarios (less quality degradation) and underweights errors on "harder" scenarios (more quality degradation) when computing averages. We refer to the difficulty normalized versions of WER/WERD as **Normalized WER/WERD** (NWER/NWERD). We estimate speech quality using DNSMOS [Reddy et al., 2019] and PESQ[Rix et al., 2001, Miao Wang and ananda seelan, 2022], which are models of human judgments of speech quality and predict Mean Opinion Scores (MOS, Rec 2018). PESQ uses various signal processing methods to predict MOS, while DNSMOS uses DNNs to do the same. To compute speech quality degradation we compute PESQ and DNSMOS for each clean and noisy recording multiplied by -1 (lower values indicate less degradation). Since we are only interested in the relative degradation between scenarios, we normalize the scores to have mean 50 and standard deviation 25.

Note on usage: We use NWERD for non-adversarial scenarios (1-5) but WERD for adversarial attacks because adversarial attacks are model-specific and thus DNSMOS/PESQ scores for adversarially perturbed audio will be different for each model, which will lead to a different normalization during NWERD computation and make comparisons difficult.

9.3 Results

We evaluate several recent ASR DNNs (§9.3.1) using `SRB` and analyze the results to uncover fine-grained differences in their robustness in various challenging scenarios. We further extend our analysis by measuring ASR model robustness for various sub-groups, namely English speech and non-English (Spanish) speech, and male and female speakers. Prior works [Liu et al., 2022, Veliche and Fung, 2023] observe that there is a disparity in transcription quality between subgroups. Our analysis augments these observations by showing that inter-group disparities in robustness may also exist, thus demonstrating the utility of `SRB` in the broader field of trustworthy AI.

9.3.1 Models

For English, we evaluate Whisper [Radford et al., 2023] large-v2, base, medium, small, and tiny (wsp-{lg,bs,md,sm,tn}), Wav2Vec-2.0 [Baevski et al., 2020] base, large, self-trained large [Xu et al., 2021], and Robust Wav2Vec [Likhomanenko et al., 2020] (w2v2-{bs,lg,lg-slf,lg-rob}), HuBERT [Hsu et al., 2021a] large and XL (hubt-{lg,xl}), Nvidia Canary [NVIDIA] (cnry-1b), Nvidia Parakeet RNN-T and CTC [NVIDIA] with 0.6B and 1.1B parameters (prkt-rnnt-{0.6,1.1}b, prkt-ctc-{0.6,1.1}b), MMS [Pratap et al., 2020] (mms-1b), Speech-T5 [Ao et al., 2022] (spch-t5), DeepSpeech [Amodei et al., 2016] (ds), and Speechbrain [Ravanelli et al., 2024] models with Conformer encoders, and transformer and RNN-T decoders. For Spanish speech, we evaluate mono-lingual Wav2Vec base Spanish [Wang et al., 2021b] (w2v2-bs-es), Wav2Vec XLSR Spanish [Conneau et al., 2020] (w2v2-lg-es), wsp-{lg,bs,tn}, and mms-1b. We used the Huggingface implementations where available, except ds (https://github.com/SeanNaren/deepspeech.pytorch). More details about the models are in Table 11.4.

Lang	Model	clean (WER)	accent	audio proc	noise (env)	noise (white)	sFX	social (FF)	social (NF)	spatial	synth speech	AVG	Adv (UA)	Adv (US)	AVG
							(NWERD)							(WERD)	
EN	prkt-rnnt-1.1b	**5.9**	11.6	**8.7**	4.2	**1.6**	6.4	48.3	39.8	11.8	**3.0**	15.0	10.9	69.1	40.0
	cnry-1b	6.0	13.9	18.2	4.4	2.7	15.0	45.7	36.7	15.3	5.7	17.5	14.8	61.7	38.3
	prkt-ctc-1.1b	6.0	16.9	10.3	4.2	3.2	9.6	44.6	35.0	13.5	5.9	15.9	8.4	71.2	39.8
	w2v2-lg-slf	7.7	41.0	30.7	13.1	17.6	20.3	69.5	67.6	26.3	15.2	33.5	7.0	41.0	24.0
	wsp-md	7.9	12.4	27.8	**2.8**	3.3	3.5	41.8	35.5	5.2	6.1	15.4	3.7	56.6	30.1
	wsp-lg	8.0	**11.2**	12.7	3.1	2.9	**2.8**	**40.9**	**34.9**	**4.5**	4.4	**13.0**	6.2	53.6	29.9
	hubt-xl	8.4	38.9	29.1	15.5	16.2	20.8	69.5	71.2	26.4	13.5	33.5	13.9	36.8	25.3
	wsp-bs	9.6	30.1	88.8	8.7	9.6	22.5	63.9	47.8	17.9	12.6	33.6	**2.7**	88.5	45.6
	w2v2-lg	9.7	60.6	39.5	19.0	26.0	24.1	77.3	79.9	37.3	17.7	42.4	16.6	**31.1**	**23.8**
ES	cnry-1b	**3.2**	699.9	21.7	17.4	7.6	30.5	-	-	36.3	28.6	120.3	26.1	84.3	55.2
	wsp-lg	5.8	**6.8**	**19.4**	**12.3**	**5.5**	**9.2**	-	-	**5.5**	**24.6**	**11.9**	13.7	65.0	39.4
	w2v2-lg-es	6.8	31.6	31.8	30.1	20.5	40.7	-	-	89.0	104.1	49.7	33.9	71.0	52.4
	wsp-bs	14.8	62.0	133.2	43.1	25.8	60.4	-	-	58.1	87.8	67.2	19.5	159.5	89.5
	mms-1b	15.7	26.3	27.9	32.7	9.3	43.3	-	-	47.3	49.9	33.8	**7.4**	53.8	30.6
	w2v2-bs-es	25.7	45.6	55.9	44.9	25.0	58.4	-	-	103.3	152.7	69.4	10.0	**33.8**	**21.9**

Table 9.1: The utility and robustness of selected English and Spanish models (see Table 11.6 for more results). Utility is measured by WER of the models on clean speech. Robustness is measured by the NWERD on non-adversarially perturbed speech and WERD on adversarially perturbed speech. Adv (UA) refers to utterance agnostic attacks, while Adv (US) refers to utterance specific ones. The metrics are averaged over all datasets, perturbations, and severities in each category.

9.3.2 Robustness of ASR Models

Table 9.1 presents the utility and robustness of a subset of English and Spanish ASR models under non-adversarial and adversarial scenarios. The subset was selected to exclude small and/or less accurate models. The results of the excluded models, however, are used in § 9.3.3.

Robustness in Non-Adversarial Scenarios

English Models: In terms of average NWERD, we observe that wsp-lg emerges as the most robust model for non-adversarial scenarios, followed by prkt-rnnt-1.1b and wsp-md. Interestingly, cnry-1b, which is the top model on the Open ASR Leaderboard (OAL, Srivastav et al. 2023), ranks 5th on SRB. This result highlights the fact that SRB provides a more rigorous assessment of a model's robustness than existing benchmarks like OAL. We also see that SRB reveals subtle weaknesses and strengths of various models.

For instance, we see that wsp-lg and wsp-md are significantly more robust to special effects (sFX) and spatial acoustics than other models, including cnry-1b. To identify the specific types of sFX and spatial acoustic perturbations against which cnry-1b lacks robustness, we plot the WERD on each perturbation within these categories (Fig. 9.3) we find that cnry-1b is much more sensitive than its peers to echo, real room impulse responses, and speed and pitch modifications. This analysis demonstrates that SRB can evaluate the robustness of ASR models at multiple granularities and can pinpoint the weaknesses of a given model. Detailed results can be found in Figs. 11.2 and 11.4 in the appendix. Given that no data augmentation, other than SpecAugment [Park et al.,

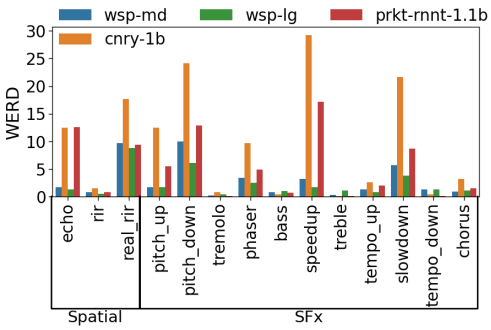


Figure 9.3: WERD of cnry-1b, wsp-md and wsp-lg on perturbations in the spatial acoustics and special effects categories.

2019], was used to train Whisper [Radford et al., 2023], this indicates that Whisper was trained on data that may have included digital media like music or movie soundtracks, and speech recorded in diverse acoustic environments – settings that may not be sufficiently represented in public data sources. Curating such diverse datasets is a promising direction for future work.

We also note that despite being pre-trained on 60K hours of speech, Wav2Vec and Hubert models severely lack robustness. Particularly concerning is their weak performance on accented speech, social scenarios and spatial acoustics, which models are very likely to encounter in the real world.

Takeaways: *(1) Despite topping the Open ASR Leaderboard,* cnry-1b *is significantly less robust than* wsp-lg*, which is ranked 10 on OAL.* cnry-1b *particularly lacks robustness to special effects and spatial acoustics. (2) Wav2Vec variants struggle against accented speech and social settings, thus, may not be suitable when users have diverse accents.*

Spanish Models: We observe that wsp-lg is the most robust model against non-adversarial perturbations by some margin. We notice that all models, except wsp-lg, struggle against accented speech and yield high NWERS. cnry-1b is particularly, weak against accented speech with an NWERD of 700% (WERD=205%). Apart from accented speech, cnry-1b is quite robust on all other categories of non-adversarial perturbations. mms-1b is also fairly robust and, unlike other models, its NWERD does not vary erratically from one category to another.

Robustness in Adversarial Scenarios

English models: We observe that w2v2-lg achieves the lowest WERD and thus is the most robust model against utterance-specific adversarial attacks. Interestingly, while Wav2Vec models exhibited mediocre robustness to non-adversarial perturbations, they are more robust to utterance-specific attacks, than Whisper, Canary, and Parakeet, which were the most robust on non-adversarial perturbations. We also note from Fig. 11.1 (in Appendix) that most Wav2Vec models are considerably more robust to attacks against TEDLIUM than against LibriSpeech, and the opposite is true for whisper and Canary models. Under utterance-agnostic attacks, the most robust models are mms-1b, wsp-bs, and wsp-sm. It is interesting to note that the smaller variants of Whisper limit the generalizability across utterances of the adversarial perturbations to a greater extent than their larger counterparts.

Takeaway: *Wav2Vec models are most robust to adversarial attacks; Models that are most robust to non-adversarial perturbations, are mediocre against adversarial perturbations; Canary and Parakeet models are highly vulnerable to utterance specific attacks.*

Spanish models: On Spanish, w2v2-bs-es is the most adversarially robust model. Generally, we observe that Wav2Vec models exhibit better robustness than Whisper and Canary under both utterance-agnostic and utterance-specific perturbations. This is similar to the trends observed in English speech (Fig. 11.1c). Detailed results can be found in Fig. 11.3 in the appendix.

Takeaway: *General trends similar to English but WERD is higher when Spanish speech is attacked.*

9.3.3 Correlates of Robustness

To glean insights that can inform future work, we have conducted the following analysis to model attributes that yield robust models. Specifically, we examine the impact of model size, architecture and accuracy, as well as training dataset size on robustness.

To determine if the prevailing practice of training DNNs with more parameters on larger datasets is yielding improvements in robustness, we use robust linear regression to fit a line to WERD vs. number of model parameters/size of the training data for the candidate models in Figs. 9.5a and 9.5b, respectively. Increasing model size is correlated with improved robustness (lower WERD).

To further isolate the impact of the model size we plot the NWERD of models from the same family in Fig. 9.4, which have similar architectures and training datasets. We note that larger models are more robust in the Whisper, Parakeet and Wav2Vec-2.0 families, but, surprisingly, not in the HuBert family.

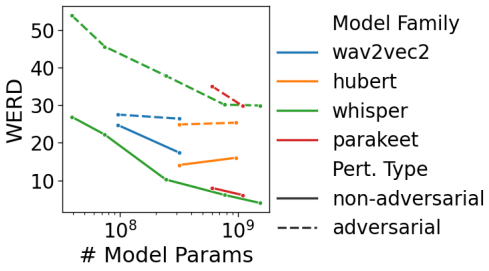


Figure 9.4: NWERD lineplot with non-adversarial and adversarial perturbations, three families of models.

Next, we consider the model architectures. The architectures of the models used in this paper can be divided in to three categories: sequence-to-sequence (seq2seq) models like Whisper and Canary, encoder only models trained with CTC loss [Graves et al., 2006] like the Wav2Vec family, and RNN-T models which are capable of streaming such as some variants of Parakeet. From Fig. 9.5e we see that in terms of non adversarial robustness RNN-T models outperform seq2seq and CTC models, but in terms of adversarial robustness CTC models achieve the lowest WERD.

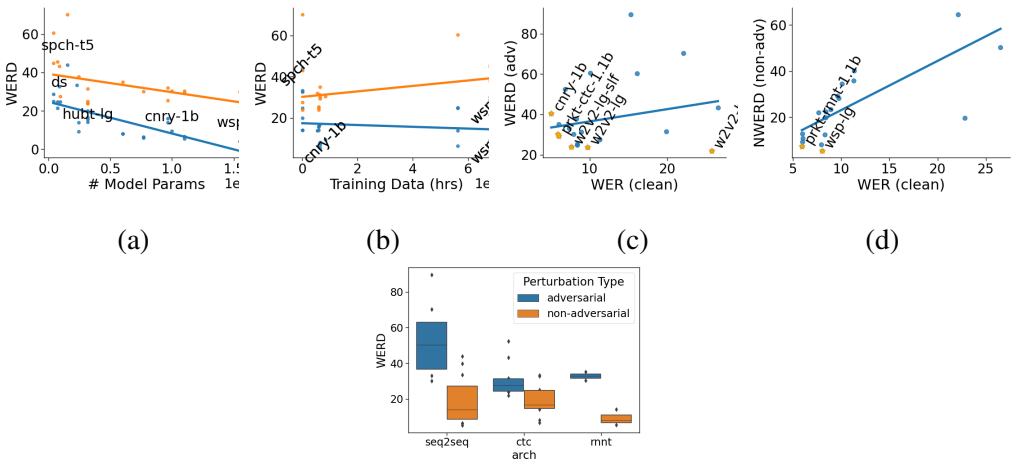
We also measure the robustness-utility trade-off by plotting WERD and NWERD for adversarial and non-adversarial perturbations, respectively, against WER on clean data in Figs. 9.5c and 9.5d. We observe that in both cases the relationship is positive, i.e. more accurate models tend to be more robust, however, the relationship between WERD on adversarial perturbations and clean WER is much weaker.

Finally, we measure the impact of training data size and, find that increasing training data appears to have only a minor influence on robustness (Fig. 9.5b).

Takeaway: *(1) Larger models tend to be more robust, while smaller models, even if they are trained on large datasets, are less robust. This runs somewhat counter to the prevailing wisdom [Radford et al., 2023, Likhomanenko et al., 2020]. (2) CTC models are more robust than seq2seq models to adversarial attack, but less robust than seq2seq and RNN-T models on non-adversarial perturbations. (3) Utility and robustness are positively correlated, but correlation is weaker for adversarial robustness.*

9.3.4 Disparity in Robustness Across Population Sub-Groups

In the preceding analysis, we considered robustness aggregated over the entire population (i.e., dataset). However, populations are generally not homogeneous, and, thus, the robustness of the model may differ on various population sub-groups. Prior works have commonly analyzed sub-group fairness of ASR models by comparing the overall WER for each sub-group on a benchmark dataset [Koe-



(a) (b) (c) (d)

(e)

Figure 9.5: (a & b) WERD for all models with robust regression fitted line on non-adversarial (blue) and adversarial (orange) perturbations, plotted against (a) number of parameters, and (b) hours of training data. (c % d) WERD and NWER on adversarial and non-adversarial perturbations are plotted against WER to illustrate the robustness-utility trade off. Pareto optimal points are highlighted. (e) Boxplot of WERD for models having various architectures.

necke et al., 2020]. It is possible that models that are fair *on average*, may not be fair under certain conditions. In the following, we use `SRB` to uncover and analyze the disparities in the models' robustness across four sub-groups: English and Spanish speech, and male and female speakers. We find that disparities indeed exist, with multi-lingual models generally being more robust for English than Spanish (Fig. 9.6), and most models being less robust for females than males.

Disparity in Robustness Across Languages in Multi-Lingual Models

We compare the robustness exhibited by multi-lingual models, wsp-lg, wsp-bs, cnry-1b and mms-1b on English and Spanish. The WERD of these models on both languages is presented in Fig. 9.6. We observe that Whisper models achieve lower WERD on English speech than on Spanish on almost all perturbation categories, while cnry-1b and mms-1b achieve similar WERD on some categories. We also note that the difference in WERD on some common perturbation categories, like environmental noise, and spatial acoustics, is much greater for wsp-lg than for cnry-1b.

Takeaway: *Multilingual models are more robust on English than Spanish;* cnry-1b *and* wsp-lg *most robust on both languages; adversarial robustness results follow the same trend as English.*

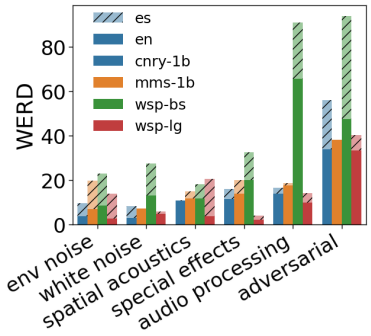


Figure 9.6: Comparing the robustness of multi-lingual on English (solid) and Spanish (hatched).

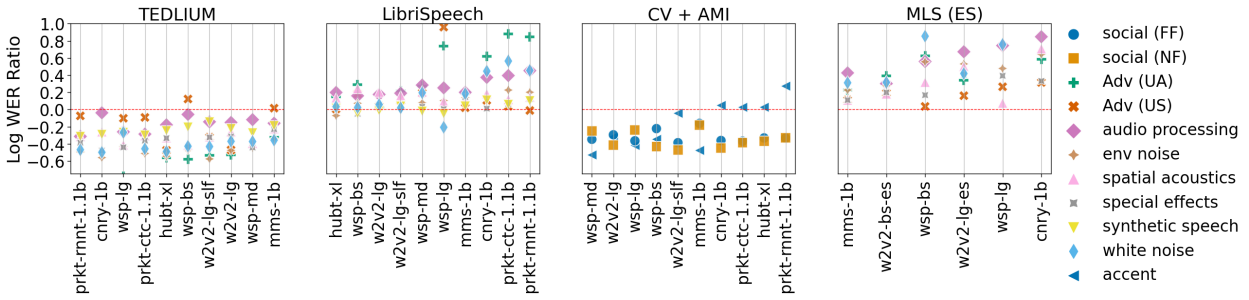


Figure 9.7: Log WER Ratio for various TEDLIUM, LibriSpeech, Common Voice (CV) and AMI, and Spanish Multilingual Librispeech (MLS (ES)). WERs are averaged across severity levels and individual augmentations within each category before computing the Log WER Ratio.

Disparity in Robustness Across Genders

To measure the disparity in transcription quality across genders (males/females), we compute the log of the ratio of the WERs of the ASR model on female and male speakers. We call this measure the Log WER Ratio (LWERR). A positive value of LWERR indicates that the model is biased against females and a negative value indicates that the model is biased against males.

The LWERR for each dataset is shown in Fig. 9.7. We note that, on average, the models are biased against females on LibriSpeech and Spanish Multilingual Librispeech (MLS-ES), and against males on TEDLIUM, Common Voice and AMI. The bias is most prominent in MLS-ES, where cnry-1b seems to be yielding the highest disparities among genders. We also note that adversarial perturbations cause the WER of wsp-lg to increase significantly more for females than males in LibriSpeech. This is interesting because adversarial perturbations do not target a specific part of the spectrum and thus should not impact one gender more than the other.

Takeaway: *Models are more robust for males on some datasets, and females on other datasets suggesting that used data require further examination; adversarial attacks increase WER of Whisper variants for females more than males; multilingual models, particularly cnry-1b, are more biased against females when transcribing Spanish.*

Chapter 10

Conclusion

In this thesis, we work towards DNNs that are robust to a variety of adversarial attacks by identifying principles, or priors, that can endow DNNs with robustness to adversarial attacks, without training them on adversarially perturbed data. In this connection, we have studied priors over the design elements and the structure of DNNs (structural priors) as well as priors derived from biological perception (biological priors) that seek to simulate biological mechanisms and constraints considered to be conducive to robustness.

The outcomes of our studies indicate that we *can* endow DNNs with a degree of generalized adversarial robustness by incorporating certain robustness priors related to the architecture and feature representations of the DNN, and without training them on a variety of adversarial attacks. This represents a step taken towards developing models that retain accuracy in the face of a variety of adversarial attacks and thus can be safely and reliably deployed in real-world settings. Our studies also reveal that deriving these priors from biology is a promising direction, and one that allows us to leverage the optimizations performed by evolution over millennia that have endowed humans and other primates with robust perception. The fact that integrating biological priors indeed endows DNNs with generalizable robustness indicates that doing so bridges some of the gaps between DNNs and biological perception, at least so far as robustness is concerned. In a word, these results provide evidence that there are more practical and scalable alternatives to the dominant approach of seeking robustness via training, and that DNNs with high accuracy and generalized adversarial robustness are, in fact, within reach. We envision that future research directions stemming from our work will further expand the body of structural and biological robustness-enhancing priors as well as discover other types of priors, particularly those related to the optimization algorithms used to learn DNN parameters.

Bibliography

W. B. *, J. R. *, and M. Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SyZI0GWCZ.

S. Addepalli, V. BS, A. Baburaj, G. Sriramanan, and R. V. Babu. Towards achieving adversarial robustness by enforcing feature consistency across bit planes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1020–1029, 2020.

N. Akhtar, A. Mian, N. Kardan, and M. Shah. Advances in adversarial attacks and defenses in computer vision: A survey. *IEEE Access*, 9:155161–155196, 2021.

D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR, 2016.

M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020.

J. Ao, R. Wang, L. Zhou, C. Wang, S. Ren, Y. Wu, S. Liu, T. Ko, Q. Li, Y. Zhang, Z. Wei, Y. Qian, J. Li, and F. Wei. SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5723–5738, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.393. URL https://aclanthology.org/2022.acl-long.393.

R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*, 2019.

A. Athalye, N. Carlini, and D. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018a.

A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018b.

A. Baevski, Y. Zhou, A. Mohamed, and M. Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33: 12449–12460, 2020.

T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021.

A. Bansal, A. Schwarzschild, E. Borgnia, Z. Emam, F. Huang, M. Goldblum, and T. Goldstein. End-to-end algorithm synthesis with recurrent networks: Extrapolation without overthinking. In *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, 2022.

J. P. Barker, R. Marxer, E. Vincent, and S. Watanabe. The chime challenges: Robust speech recognition in everyday environments. *New era for robust speech recognition: Exploiting deep learning*, pages 327–344, 2017.

M. I. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. D. Hjelm. Mine: mutual information neural estimation. *arXiv:1801.04062*, 2018.

P. Benz, C. Zhang, and I. S. Kweon. Batch normalization increases adversarial vulnerability and decreases adversarial transferability: A non-robust feature perspective. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7818–7827, 2021.

F. Briggs. Role of feedback connections in central visual processing. *Annual Review of Vision Science*, 6:313–334, 2020.

S. Bubeck and M. Sellke. A universal law of robustness via isoperimetry. *Advances in Neural Information Processing Systems*, 34:28811–28822, 2021.

J. Bullier, J.-M. Hupé, A. C. James, and P. Girard. Chapter 13 the role of feedback connections in shaping the responses of visual cortical neurons. In *Vision: From Neurons to Cognition*, volume 134 of *Progress in Brain Research*, pages 193–204. Elsevier, 2001. doi: https://doi.org/10.1016/S0079-6123(01)34014-1. URL https://www.sciencedirect.com/science/article/pii/S0079612301340141.

C. F. Cadieu, H. Hong, D. L. Yamins, N. Pinto, D. Ardila, E. A. Solomon, N. J. Majaj, and J. J. DiCarlo. Deep neural networks rival the representation of primate it cortex for core visual object recognition. *PLoS computational biology*, 10(12):e1003963, 2014.

X. Cao and N. Z. Gong. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pages 278–287, 2017.

N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

N. Carlini, F. Tramer, J. Z. Kolter, et al. (certified!!) adversarial robustness for free! *arXiv preprint arXiv:2206.10550*, 2022.

E. Casanova, J. Weber, C. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone.(12 2021), 2021.

E. Casanova, J. Weber, C. D. Shulby, A. C. Junior, E. Gölge, and M. A. Ponti. Yourtts: Towards zero-shot multi-speaker tts and zero-shot voice conversion for everyone. In *International Conference on Machine Learning*, pages 2709–2720. PMLR, 2022.

A. Chan, Y. Tay, Y. S. Ong, and J. Fu. Jacobian adversarially regularized networks for robustness. *arXiv preprint arXiv:1912.10185*, 2019.

C. Chen, N. Hou, Y. Hu, S. Shirol, and E. S. Chng. Noise-robust speech recognition with 10 minutes unparalleled in-domain data. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4298–4302. IEEE, 2022.

J. Chen, M. I. Jordan, and M. J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pages 1277–1294. IEEE, 2020a.

P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li. Continuous speech separation: Dataset and analysis. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7284–7288. IEEE, 2020b.

B. Choksi, M. Mozafari, C. Biggs O'May, B. Ador, A. Alamia, and R. VanRullen. Predify: Augmenting deep neural networks with brain-inspired predictive coding dynamics. *Advances in Neural Information Processing Systems*, 34:14069–14083, 2021.

J. Cohen, E. Rosenfeld, and Z. Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019a.

J. M. Cohen, E. Rosenfeld, and J. Z. Kolter. Certified adversarial robustness via randomized smoothing. *CoRR*, 2019b.

A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*, 2020.

J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent. Librimix: An open-source dataset for generalizable speech separation. *arXiv preprint arXiv:2005.11262*, 2020.

F. Croce and M. Hein. Provable robustness against all adversarial l_p-perturbations for $p \geq 1$. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=rklk_ySYPB.

F. Croce and M. Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR, 2020b.

F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020c.

F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

L. J. Croner, K. Purpura, and E. Kaplan. Response variability in retinal ganglion cells of primates. *Proceedings of the National Academy of Sciences*, 90(17):8128–8130, 1993.

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

A. Dabouei, S. Soleymani, F. Taherkhani, J. Dawson, and N. M. Nasrabadi. Exploiting joint robustness to adversarial perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1122–1131, 2020.

J. Dapello, T. Marques, M. Schrimpf, F. Geiger, D. Cox, and J. J. DiCarlo. Simulating a primary visual cortex at the front of cnns improves robustness to image perturbations. *Advances in Neural Information Processing Systems*, 33:13073–13087, 2020.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

G. W. Ding, L. Wang, and X. Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv:1902.07623*, 2019.

S. Dodge and L. Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)*, pages 1–7. IEEE, 2017.

Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

V. Dragoi and C. Tsuchitani. Visual processing: Eye and retina (section 2, chapter 14) neuroscience online: An electronic textbook for the neurosciences: Department of neurobiology and anatomy - the university of texas medical school at houston, 2020. URL https://nba.uth.tmc.edu/neuroscience/m/s2/chapter14.html.

N. R. K. Elena Rastorgueva. New Standard for Speech Recognition and Translation from the NVIDIA NeMo Canary Model. URL https://developer.nvidia.com/blog/new-standard-for-speech-recognition-and-translation-from-the-nvidia-nemo-canary-model/.

J. Feather, A. Durango, R. Gonzalez, and J. McDermott. Metamers of neural networks reveal divergence from human perceptual systems. In *Advances in Neural Information Processing Systems*, pages 10078–10089, 2019.

J. Feather, G. Leclerc, A. Mądry, and J. H. McDermott. Model metamers reveal divergent invariances between biological and artificial neural networks. *Nature Neuroscience*, pages 1–18, 2023.

M. Fischer, M. Baader, and M. Vechev. Certified defense to image transformations via randomized smoothing. *Advances in Neural Information Processing Systems*, 33:8404–8417, 2020.

Y. Fu, Q. Yu, Y. Zhang, S. Wu, X. Ouyang, D. Cox, and Y. Lin. Drawing robust scratch tickets: Subnetworks with inborn robustness are found within randomly initialized networks. *Advances in Neural Information Processing Systems*, 34:13059–13072, 2021.

J. M. Gant, A. Banburski, and A. Deza. Evaluating the adversarial robustness of a foveated texture transform module in a cnn. In *SVRHM 2021 Workshop@ NeurIPS*, 2021.

R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018a.

R. Geirhos, C. R. Temme, J. Rauber, H. H. Schütt, M. Bethge, and F. A. Wichmann. Generalisation in humans and deep neural networks. *Advances in neural information processing systems*, 31, 2018b.

R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

O. Ghitza. Auditory nerve representation as a front-end for speech recognition in a noisy environment. *Computer Speech & Language*, 1(2):109–130, 1986.

J. J. Godfrey, E. C. Holliman, and J. McDaniel. Switchboard: Telephone speech corpus for research and development. In *Acoustics, speech, and signal processing, ieee international conference on*, volume 1, pages 517–520. IEEE Computer Society, 1992.

M. D. Golub, P. T. Sadtler, E. R. Oby, K. M. Quick, S. I. Ryu, E. C. Tyler-Kabara, A. P. Batista, S. M. Chase, and B. M. Yu. Learning by neural reassociation. *Nature neuroscience*, 2018.

I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples, 2014.

A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

M. Guo, Y. Yang, R. Xu, Z. Liu, and D. Lin. When nas meets robustness: In search of robust architectures against adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 631–640, 2020.

Y. Guo, C. Zhang, C. Zhang, and Y. Chen. Sparse dnns with improved adversarial robustness. In *NeurIPS*, 2018.

T. Hansen, L. Pracejus, and K. R. Gegenfurtner. Color perception in the intermediate periphery of the visual field. *Journal of vision*, 9(4):26–26, 2009.

A. Harrington and A. Deza. Finding biological plausibility for adversarially robust features via metameric tasks. In *International Conference on Learning Representations*, 2021.

D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021a.

D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021b.

J. A. Hennig, E. R. Oby, D. M. Losey, A. P. Batista, M. Y. Byron, and S. M. Chase. How learning unfolds in the brain: toward an optimization view. *Neuron*, 2021.

H. Hermansky, N. Morgan, A. Bayya, and P. Kohn. Rasta-plp speech analysis. In *Proc. IEEE Int'l Conf. Acoustics, speech and signal processing*, volume 1, pages 121–124, 1991.

F. Hernandez, V. Nguyen, S. Ghannay, N. Tomashenko, and Y. Esteve. Ted-lium 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20*, pages 198–208. Springer, 2018.

J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 31–35. IEEE, 2016.

G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015.

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

L. Hsiung, Y.-Y. Tsai, P.-Y. Chen, and T.-Y. Ho. Towards compositional adversarial robustness: Generalizing adversarial training to composite semantic perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24658–24667, 2023.

W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021a.

W.-N. Hsu, A. Sriram, A. Baevski, T. Likhomanenko, Q. Xu, V. Pratap, J. Kahn, A. Lee, R. Collobert, G. Synnaeve, et al. Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training. *arXiv preprint arXiv:2104.01027*, 2021b.

H. Huang, Y. Wang, S. Erfani, Q. Gu, J. Bailey, and X. Ma. Exploring architectural ingredients of adversarially robust deep neural networks. *Advances in Neural Information Processing Systems*, 34:5545–5559, 2021.

S. Huang, Z. Lu, K. Deb, and V. N. Boddeti. Revisiting residual networks for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8202–8211, 2023.

Y. Huang, J. Gornet, S. Dai, Z. Yu, T. Nguyen, D. Tsao, and A. Anandkumar. Neural networks with recurrent generative feedback. *Advances in Neural Information Processing Systems*, 33: 535–545, 2020.

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*, 32, 2019.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ioffe15.html.

M. Jeub, M. Schafer, and P. Vary. A binaural room impulse response database for the evaluation of dereverberation algorithms. In *2009 16th International Conference on Digital Signal Processing*, pages 1–5. IEEE, 2009.

A. Jonnalagadda, W. Y. Wang, B. Manjunath, and M. Eckstein. Foveater: Foveated transformer for image classification, 2022. URL https://openreview.net/forum?id=mqIeP6qPvta.

S. Joos, T. Van hamme, D. Preuveneers, and W. Joosen. Adversarial robustness is not enough: Practical limitations for securing facial authentication. In *Proceedings of the 2022 ACM on International Workshop on Security and Privacy Analytics*, pages 2–12, 2022.

J. L. Julia Angwin. Machine Bias. URL https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing.

D. Kang, Y. Sun, D. Hendrycks, T. Brown, and J. Steinhardt. Testing robustness against unforeseen adversaries. *arXiv preprint arXiv:1908.08016*, 2019.

C. Kim and R. M. Stern. Power-normalized cepstral coefficients (pncc) for robust speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 24(7):1315–1329, 2016.

K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas, et al. The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4. IEEE, 2013.

T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5220–5224. IEEE, 2017.

A. Koenecke, A. Nam, E. Lake, J. Nudell, M. Quartey, Z. Mengesha, C. Toups, J. R. Rickford, D. Jurafsky, and S. Goel. Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences*, 117(14):7684–7689, 2020.

H. Kolb. Facts and figures concerning the human retina - ncbi bookshelf, May 2005. URL https://www.ncbi.nlm.nih.gov/books/NBK11556/.

W. Kraaij, T. Hain, M. Lincoln, and W. Post. The ami meeting corpus. In *Proc. International Conference on Methods and Techniques in Behavioral Research*, 2005.

A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

J. Kubilius, M. Schrimpf, A. Nayebi, D. Bear, D. L. Yamins, and J. J. DiCarlo. Cornet: Modeling the neural mechanisms of core object recognition. *BioRxiv*, page 408385, 2018.

A. Kumar and T. Goldstein. Center smoothing: Certified robustness for networks with structured outputs. *Advances in Neural Information Processing Systems*, 34:5560–5575, 2021.

C. Laidlaw and S. Feizi. Functional adversarial attacks. *Advances in neural information processing systems*, 32, 2019.

C. Laidlaw, S. Singla, and S. Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=dFwBosAcJkN.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2010.

C. Lenk, P. Hövel, K. Ved, S. Durstewitz, T. Meurer, T. Fritsch, A. Männchen, J. Küller, D. Beer, T. Ivanov, et al. Neuromorphic acoustic sensing using an adaptive microelectromechanical cochlea with integrated feedback. *Nature Electronics*, pages 1–11, 2023.

B. Li, C. Chen, W. Wang, and L. Carin. Certified adversarial robustness with additive noise. *Advances in neural information processing systems*, 32, 2019.

T. Likhomanenko, Q. Xu, V. Pratap, P. Tomasello, J. Kahn, G. Avidov, R. Collobert, and G. Synnaeve. Rethinking evaluation in asr: Are our models robust enough? *arXiv preprint arXiv:2010.11745*, 2020.

Y. Lin, W. H. Abdulla, Y. Lin, and W. H. Abdulla. Principles of psychoacoustics. *Audio Watermark: A Comprehensive Foundation Using MATLAB*, pages 15–49, 2015.

C. Liu, M. Picheny, L. Sarı, P. Chitkara, A. Xiao, X. Zhang, M. Chou, A. Alvarado, C. Hazirbas, and Y. Saraf. Towards measuring fairness in speech recognition: Casual conversations dataset transcriptions. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6162–6166. IEEE, 2022.

N. Loo, R. Hasani, A. Amini, and D. Rus. Evolution of neural tangent kernels under benign and adversarial training. *Advances in Neural Information Processing Systems*, 35:11642–11657, 2022.

Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao. Foveation-based mechanisms alleviate adversarial examples. *arXiv preprint arXiv:1511.06292*, 2015.

R. Lyon. Computational models of neural auditory processing. In *ICASSP'84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 41–44. IEEE, 1984.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks, 2017.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018a. URL https://openreview.net/forum?id=rJzIBfZAb.

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*, 2018b.

P. Maini, E. Wong, and Z. Kolter. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning*, pages 6640–6650. PMLR, 2020.

N. T. Markov, J. Vezoli, P. Chameau, A. Falchier, R. Quilodran, C. Huissoud, C. Lamy, P. Misery, P. Giroud, S. Ullman, et al. Anatomy of hierarchy: feedforward and feedback pathways in macaque visual cortex. *Journal of Comparative Neurology*, 522(1):225–259, 2014.

J. Mehrer, C. J. Spoerer, E. C. Jones, N. Kriegeskorte, and T. C. Kietzmann. An ecologically motivated image dataset for deep learning yields better models of human vision. *Proceedings of the National Academy of Sciences*, 118(8):e2011417118, 2021.

J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. On detecting adversarial perturbations. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=SJzCSf9xg.

R. G. D. Miao Wang, Christoph Boeddeker and ananda seelan. Pesq (perceptual evaluation of speech quality) wrapper for python users, May 2022. URL https://doi.org/10.5281/zenodo.6549559.

S. Nakamura, K. Hiyane, F. Asano, T. Nishiura, and T. Yamada. Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition. In *LREC*, pages 965–968, 2000.

V. Nanda, S. Dooley, S. Singla, S. Feizi, and J. P. Dickerson. Fairness through robustness: Investigating robustness disparity in deep learning. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 466–477, 2021.

P. Neekhara, S. Hussain, P. Pandey, S. Dubnov, J. McAuley, and F. Koushanfar. Universal adversarial perturbations for speech recognition systems. *arXiv preprint arXiv:1905.03828*, 2019.

NVIDIA. NeMo Documentation. https://docs.nvidia.com/nemo-framework/user-guide/latest/nemotoolkit/asr/models.html.

R. Olivier and B. Raj. Recent improvements of asr models in the face of adversarial attacks. *Interspeech*, 2022a. URL https://arxiv.org/abs/2203.16536.

R. Olivier and B. Raj. Recent improvements of asr models in the face of adversarial attacks. *arXiv preprint arXiv:2203.16536*, 2022b.

D. M. Paiton, C. G. Frye, S. Y. Lundquist, J. D. Bowen, R. Zarcone, and B. A. Olshausen. Selectivity and robustness of sparse coding networks. *Journal of vision*, 20(12):10–10, 2020.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

N. Papernot, P. McDaniel, and I. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.

A. Patane, A. Blaas, L. Laurenti, L. Cardelli, S. Roberts, and M. Kwiatkowska. Adversarial robustness guarantees for gaussian processes. *The Journal of Machine Learning Research*, 23(1): 6524–6578, 2022.

Y. Peng, S. Dalmia, I. Lane, and S. Watanabe. Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding. In *International Conference on Machine Learning*, pages 17627–17643. PMLR, 2022.

K. J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press, 2015. ISBN 978-1-4503-3459-4. doi: 10.1145/2733373.2806390. URL http://dl.acm.org/citation.cfm?doid=2733373.2806390.

V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*, 2020.

V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi, et al. Scaling speech technology to 1,000+ languages. *arXiv preprint arXiv:2305.13516*, 2023.

Y. Qin, N. Carlini, G. Cottrell, I. Goodfellow, and C. Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International conference on machine learning*, pages 5231–5240. PMLR, 2019.

A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever. Robust speech recognition via large-scale weak supervision. In *International Conference on Machine Learning*, pages 28492–28518. PMLR, 2023.

E. Raff, J. Sylvester, S. Forsyth, and M. McLean. Barrage of random transforms for adversarially robust defense. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6528–6537, 2019.

F. Ramezani, S. R. Kheradpisheh, S. J. Thorpe, and M. Ghodrati. Object categorization in visual periphery is modulated by delayed foveal noise. *Journal of Vision*, 19(9):1–1, 2019.

R. P. Rao and D. H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 1999.

M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio. SpeechBrain: A general-purpose speech toolkit, 2021. arXiv:2106.04624.

M. Ravanelli, T. Parcollet, A. Moumen, S. de Langen, C. Subakan, P. Plantinga, Y. Wang, P. Mousavi, L. D. Libera, A. Ploujnikov, F. Paissan, D. Borra, S. Zaiem, Z. Zhao, S. Zhang, G. Karakasidis, S.-L. Yeh, P. Champion, A. Rouhe, R. Braun, F. Mai, J. Zuluaga-Gomez, S. M. Mousavi, A. Nautsch, X. Liu, S. Sagar, J. Duret, S. Mdhaffar, G. Laperriere, M. Rouvier, R. D. Mori, and Y. Esteve. Open-source conversational ai with SpeechBrain 1.0, 2024. URL https://arxiv.org/abs/2407.00463.

S.-A. Rebuffi, S. Gowal, D. A. Calian, F. Stimberg, O. Wiles, and T. A. Mann. Data augmentation can improve robustness. *Advances in Neural Information Processing Systems*, 34:29935–29948, 2021.

I. Rec. P. 808, subjective evaluation of speech quality with a crowdsourcing approach. *ITU-T, Geneva*, 2018.

C. K. Reddy, E. Beyrami, J. Pool, R. Cutler, S. Srinivasan, and J. Gehrke. A scalable noisy speech dataset and online subjective test framework. *Proc. Interspeech 2019*, pages 1816–1820, 2019.

C. K. Reddy, V. Gopal, R. Cutler, E. Beyrami, R. Cheng, H. Dubey, S. Matusevych, R. Aichner, A. Aazami, S. Braun, et al. The interspeech 2020 deep noise suppression challenge: Datasets, subjective testing framework, and challenge results. In *INTERSPEECH*, 2020.

A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra. Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE international conference on acoustics, speech, and signal processing. Proceedings (Cat. No. 01CH37221)*, volume 2, pages 749–752. IEEE, 2001.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

P. T. Sadtler, K. M. Quick, M. D. Golub, S. M. Chase, S. I. Ryu, E. C. Tyler-Kabara, B. M. Yu, and A. P. Batista. Neural constraints on learning. *Nature*, 2014.

H. Salman, M. Sun, G. Yang, A. Kapoor, and J. Z. Kolter. Denoised smoothing: A provable defense for pretrained classifiers. *Advances in Neural Information Processing Systems*, 33: 21945–21957, 2020.

J. F. Santos and T. H. Falk. Updating the srmr-ci metric for improved intelligibility prediction for cochlear implant users. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 22(12):2197–2206, 2014.

L. Schott, J. Rauber, M. Bethge, and W. Brendel. Towards the first adversarially robust neural network model on mnist. *arXiv preprint arXiv:1805.09190*, 2018.

A. Schwarzschild, E. Borgnia, A. Gupta, F. Huang, U. Vishkin, M. Goldblum, and T. Goldstein. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. *Advances in Neural Information Processing Systems*, 34:6695–6706, 2021.

S. Seneff. A joint synchrony/mean-rate model of auditory speech processing. *Journal of phonetics*, 16(1):55–76, 1988.

M. Shah and B. Raj. Deriving compact feature representations via annealed contraction. In *ICASSP*, 2020.

M. Shah, R. Olivier, and B. Raj. Exploiting non-linear redundancy for neural model compression. In *ICPR*, 2021.

M. A. Shah, D. S. Noguero, M. A. Heikkilä, B. Raj, and N. Kourtellis. Speech robust bench: A robustness benchmark for speech recognition. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=D0LuQNZfEl.

A. Sharma, Y. Bian, P. Munz, and A. Narayan. Adversarial patch attacks and defences in vision-based tasks: A survey. *arXiv preprint arXiv:2206.08304*, 2022.

Y. Sharma and P.-Y. Chen. Attacking the madry defense model with l_1-based adversarial examples. *arXiv preprint arXiv:1710.10733*, 2017.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.

C. Sitawarin, Z. J. Golan-Strieb, and D. Wagner. Demystifying the adversarial robustness of random transformation defenses. In *International Conference on Machine Learning*, pages 20232–20252. PMLR, 2022.

M. Slaney. *Lyon's cochlear model*, volume 13. Citeseer, 1988.

D. Snyder, G. Chen, and D. Povey. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*, 2015.

D. Solans Noguero, D. Ramírez-Cifuentes, E. A. Ríssola, and A. Freire. Gender bias when using artificial intelligence to assess anorexia nervosa on social media: data-driven study. *Journal of Medical Internet Research*, 25:e45184, 2023.

C. Song, K. He, L. Wang, and J. E. Hopcroft. Improving the generalization of adversarial training with domain adaptation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SyfIfnC5Ym.

C. J. Spoerer, P. McClure, and N. Kriegeskorte. Recurrent convolutional neural networks: a better model of biological object recognition. *Frontiers in psychology*, 8:1551, 2017.

V. Srivastav, S. Majumdar, N. Koluguri, A. Moumen, S. Gandhi, et al. Open automatic speech recognition leaderboard. https://huggingface.co/spaces/hf-audio/open_asr_leaderboard, 2023.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15 (56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *CoRR*, abs/1505.00387, 2015. URL http://arxiv.org/abs/1505.00387.

R. M. Stern and N. Morgan. Hearing is believing: Biologically inspired methods for robust automatic speech recognition. *IEEE signal processing magazine*, 29(6):34–43, 2012.

E. E. M. Stewart, M. Valsecchi, and A. C. Schütz. A review of interactions between peripheral and foveal vision. *Journal of Vision*, 20(12):2–2, 11 2020. ISSN 1534-7362. doi: 10.1167/jov.20.12.2. URL https://doi.org/10.1167/jov.20.12.2.

Suno. Bark speaker library (v2). https://suno-ai.notion.site/8b8e8749ed514b0cbf3f699013548683?v=bc67cff786b04b50b3ceb756fd05f68c, a. Accessed: 2024-11-23.

Suno. Bark. https://github.com/suno-ai/bark, b. Accessed: 2024-11-23.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. URL http://arxiv.org/abs/1312.6199.

J. Uesato, B. O'donoghue, P. Kohli, and A. Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018.

I.-E. Veliche and P. Fung. Improving fairness and robustness in end-to-end speech recognition through unsupervised clustering. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

V. Q. Vo, E. Abbasnejad, and D. C. Ranasinghe. Query efficient decision based sparse attacks against black-box deep learning models. *arXiv preprint arXiv:2202.00091*, 2022.

M. R. Vuyyuru, A. Banburski, N. Pant, and T. Poggio. Biologically inspired mechanisms for adversarial robustness. *Advances in Neural Information Processing Systems*, 33:2135–2146, 2020.

B. Wang, C. Xu, S. Wang, Z. Gan, Y. Cheng, J. Gao, A. H. Awadallah, and B. Li. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. *arXiv preprint arXiv:2111.02840*, 2021a.

C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux. Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. *arXiv preprint arXiv:2101.00390*, 2021b.

C. Wang, M. Zhang, J. Zhao, and X. Kuang. Black-box adversarial attacks on deep neural networks: A survey. In *2022 4th International Conference on Data Intelligence and Security (ICDIS)*, pages 88–93. IEEE, 2022a.

H. Wang, X. Wu, P. Yin, and E. P. Xing. High frequency component helps explain the generalization of convolutional neural networks. *CoRR*, 2019.

H. Wang, X. Wu, Z. Huang, and E. P. Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8684–8694, 2020.

J. Wang, R. Jia, G. Friedland, B. Li, and C. Spanos. One bit matters: Understanding adversarial examples as the abuse of redundancy. *arXiv:1810.09650*, 2018.

X. Wang, H. Wang, and D. Yang. Measure and improve robustness in nlp models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, 2022b.

P. Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. In *NIPS*, 2016.

G. Wichern, J. Antognini, M. Flynn, L. R. Zhu, E. McQuinn, D. Crow, E. Manilow, and J. L. Roux. Wham!: Extending speech separation to noisy environments. *arXiv preprint arXiv:1907.01160*, 2019.

M. Wicker, X. Huang, and M. Kwiatkowska. Feature-guided black-box safety testing of deep neural networks. In *Tools and Algorithms for the Construction and Analysis of Systems: 24th International Conference, TACAS 2018, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings, Part I 24*, pages 408–426. Springer, 2018.

E. Wong and Z. Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pages 5286–5295. PMLR, 2018.

E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2019a.

E. Wong, F. Schmidt, and Z. Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *International Conference on Machine Learning*, pages 6808–6817. PMLR, 2019b.

K. Wu, A. Wang, and Y. Yu. Stronger and faster wasserstein adversarial attacks. In *International Conference on Machine Learning*, pages 10377–10387. PMLR, 2020.

D. Wyatte, D. J. Jilk, and R. C. O'Reilly. Early recurrent feedback facilitates visual object recognition under challenging conditions. *Frontiers in psychology*, 5:674, 2014.

C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HyydRMZC-.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

K. Xu, Y. Xiao, Z. Zheng, K. Cai, and R. Nevatia. Patchzero: Defending against adversarial patch attacks by detecting and zeroing the patch. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4632–4641, 2023.

Q. Xu, A. Baevski, T. Likhomanenko, P. Tomasello, A. Conneau, R. Collobert, G. Synnaeve, and M. Auli. Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3030–3034. IEEE, 2021.

J. Yamagishi, C. Veaux, and K. MacDonald. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92), 2019.

D. L. Yamins, H. Hong, C. F. Cadieu, E. A. Solomon, D. Seibert, and J. J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the national academy of sciences*, 111(23):8619–8624, 2014.

S. Zagoruyko and N. Komodakis. Wide residual networks. *British Machine Vision Conference*, 2016.

H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

P. Zhao, P.-Y. Chen, S. Wang, and X. Lin. Towards query-efficient black-box adversary with zeroth-order natural gradient descent. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6909–6916, 2020.

Z. Zhu, F. Liu, G. Chrysos, and V. Cevher. Robustness in deep learning: The good (width), the bad (depth), and the ugly (initialization). *Advances in Neural Information Processing Systems*, 35: 36094–36107, 2022.

Part IV

Appendices

Chapter 11

Robustness Benchmark For Speech Recognition Models

11.1 Perturbation Generation/Application Procedure

Below, we provide further details the perturbations that make up `Speech Robust Bench`. Table 11.1 shows the parameters for each perturbation and Table 11.2 shows the normalized DNS-MOS and PESQ scores for each perturbation.

Gaussian Noise: A noise vector of the same length as the audio signal is sample from a standard normal distribution, scaled such that its magnitude corresponds to a specific SNR, and then added to the audio signal. We use `torchaudio.function.add_noise` to add the noise to the speech at a given SNR.

Environmental Noise: We use the recordings of environmental noises from the test/eval subsets of ESC-50 [Piczak, 2015], MS-SNSD [Reddy et al., 2019], MUSAN [Snyder et al., 2015] and WHAM [Wichern et al., 2019]. We create a separate perturbed version of the clean data using each of these noise datasets. To do so, for each test utterance we sample a random environmental noise and add it to the audio signal at the specified SNR. We clip the noise if it is longer than the speech, and repeat it if it is shorter than the speech. We use `torchaudio.function.add_noise` to add the noise to the speech at a given SNR.

Room Impulse Response: The simulated and real RIRs from [Ko et al., 2017] are applied to clean recordings. As a measure of intensity, RT60 is estimated for the simulated RIRs using Sabine's formula with the room dimensions and absorption coefficient provided in the dataset. For the real RIRs, we compute the SRMR [Santos and Falk, 2014] using the implementation from https://github.com/aliutkus/speechmetrics/tree/master. The severity is defined in increasing RT60s for the synthetic RIRs, and decreasing SRMR for the real RIRs. Table 11.1 shows the average RT60/SRMS in each severity level. During evaluation, a random RIR having the given severity level is sampled for each test recording.

Category	Perturbation	Sev 1	Sev 2	Sev 3	Sev 4
Environment	Gaussian Noise	30 dB	20 dB	10 dB	0 dB
	Environmental Noise	30 dB	20 dB	10 dB	0 dB
	Music	30 dB	20 dB	10 dB	0 dB
	Crosstalk	30 dB	20 dB	10 dB	0 dB
Spatial Acoustics	RIR	0.27s	0.58s	0.99s	1.33s
	Real RIR	9.1	7.1	4.1	1.8
	Echo (delay)	125 ms	250 ms	500 ms	1000 ms
Special Effects	Bass (gain)	20	30	40	50
	Treble (gain)	10	23	36	50
	Phaser (decay)	0.3 s	0.5 s	0.7 s	0.9 s
	tempo-up	1.25x	1.5x	1.75x	2x
	tempo-down	0.875x	0.75	0.625x	0.5x
	Speed-up	1.25x	1.5x	1.75x	2x
	Slow-down	0.875x	0.75	0.625x	0.5x
	Pitch Step-up	0.25 oct	0.5 oct	0.75 oct	1 oct
	Pitch Step-down	0.25 oct	0.5 oct	0.75 oct	1 oct
	Chorus (delay)	30	50	70	90
	tremolo (depth)	50	66	83	100
Audio Processing	Resampling	0.75x	0.5x	0.25x	0.125x
	Gain (factor)	10x	20x	30x	40x
	Low-pass filter	4 kHz	2833 kHz	1666 kHz	500 kHz
	High-pass filter	500 kHz	1333 kHz	2166 kHz	3000 kHz
Adversarial	PGD Attack	40 dB	30 dB	20 dB	10dB
	Utterance Agnostic Attack	40 dB	30 dB	20 dB	10dB

Table 11.1: The parameters defining the various severity levels of the perturbations used in the proposed benchmark.

Resampling, Speed, Pitch, and Gain Perturbations: The resampling speed, pitch, and gain perturbations were applied using the `Resample Speed`, `PitchShift` and `Vol` transforms from `torchaudio`.

Other special effects: These effects are applied via SoX filters of the same name. We used `torchaudio.sox_effects.apply_effects_tensor` to apply these filters to the audio. The args for each filter are as follows:

- `echo 0.8 0.9 <delay> 0.3`
- `phaser 0.6 0.8 3 <decay> 2 "-t"`
- `Tempo <factor> 30`
- `sinc <lo-freq>`

Metric → Scenario	AVG				normalized DNSMOS				normalized PESQ			
clean	23.1				23.1							
accent (en)	33.1				33.1							
accent (es)	29.3				29.3							
social (chime, FF)	102.2				102.2							
social (ami, FF)	85.9				85.9							
social (chime, NF)	80.1				80.1							
social (ami, NF)	37.3				37.3							
Augmentation/Severity	1	2	3	4	1	2	3	4	1	2	3	4
bass	19.1	23.9	36.0	56.4	23.9	28.9	39.6	58.4	11.4	14.3	30.3	54.6
chorus	40.1	49.3	55.5	57.0	31.3	41.3	48.7	49.9	63.8	71.2	74.3	75.8
crosstalk	22.9	38.9	53.1	59.9	24.7	33.1	38.2	41.0	22.2	46.1	70.1	81.6
echo	54.9	54.2	53.6	51.4	40.6	37.8	37.6	36.5	71.3	72.7	71.7	67.9
env noise (MS-SNSD)	51.4	62.4	77.0	89.6	53.5	61.0	74.6	93.5	39.6	58.7	76.7	84.2
env noise (ESC50)	26.7	41.7	58.4	73.8	39.2	45.3	55.0	73.2	20.0	43.1	66.0	79.8
env noise (MUSAN)	24.9	43.0	63.0	76.4	26.3	38.7	57.1	73.1	24.8	48.7	70.4	81.0
env noise (WHAM)	23.0	46.2	74.2	93.3	23.7	41.2	72.3	101.7	23.1	52.0	76.6	85.4
gain	50.8	69.9	77.6	81.8	46.7	67.8	78.3	84.6	61.3	76.0	80.0	81.7
gaussian noise	53.2	76.6	91.7	82.7	72.1	89.5	103.8	119.6	42.1	69.2	83.0	66.0
highpass	40.9	56.2	68.5	78.5	35.8	45.1	66.7	83.0	46.1	68.3	71.6	74.9
lowpass	33.8	37.9	51.6	79.1	48.9	48.4	64.3	100.1	20.3	29.2	40.4	58.4
music	22.9	43.8	66.7	79.9	26.9	43.0	64.0	78.6	20.0	45.3	70.1	81.9
phaser	15.6	32.9	60.7	80.6	22.8	36.2	59.8	78.6	10.6	31.9	63.6	83.5
pitch down	61.8	68.2	63.8	84.4	39.6	50.9	67.3	82.8	85.9	86.5	68.1	86.7
pitch up	58.9	62.1	65.2	66.0	33.7	39.6	45.7	47.5	85.9	86.4	86.5	86.4
real rir	39.4	54.6	69.9	85.3	35.7	46.5	61.7	89.2	43.1	62.7	78.0	81.4
resample	14.9	28.0	49.9	64.2	25.2	43.7	63.3	77.4	6.7	18.0	38.1	52.5
rir	51.1	64.0	69.3	68.9	42.0	58.1	66.2	66.5	65.0	74.6	78.1	78.1
slowdown	51.4	57.8	65.2	74.0	18.9	31.2	45.3	68.6	85.9	86.1	86.0	86.3
speedup	52.2	59.6	67.2	73.8	23.1	37.6	52.6	65.5	83.7	83.6	83.3	83.0
tempo down	49.4	52.6	55.3	51.2	19.8	23.0	28.2	36.7	81.4	84.7	85.3	85.6
tempo up	51.0	57.9	64.1	70.6	25.7	36.1	47.8	60.1	79.0	82.2	82.4	82.4
treble	12.4	22.4	41.6	63.6	20.8	31.2	44.8	62.3	2.1	12.1	42.2	72.5
tremolo	17.7	29.9	60.2	100.2	24.5	38.8	73.9	113.7	9.6	17.6	36.9	75.6
synthetic (es, Bark)	30.7	-	-	-	30.7	-	-	-	-	-	-	-
synthetic (en, yourTTS)	50.3	-	-	-	17.1	-	-	-	83.6	-	-	-

Table 11.2: Normalized DNSMOS and PESQ score for each perturbation.

- `sinc 0-<hi-freq>`
- `tremolo 20 <depth>`

Name	Subset	Hours	Utterances	Speakers	Male/Female
LibriSpeech	test-clean	5.4	2620	40	20/20
TEDLIUM 3	test	3.76	1155	16	10/6
Multi-Lingual LibriSpeech (es)	test	10	2385	20	10/10
CHiME-6	eval	5.25	13000	8	-
AMI	test	7.35	13168	16	8/8
ESC-50	-	2.78	2000	-	-
MUSAN	-	108.5	2016	-	-
WHAM!	noise-test	9	3000	-	-
MS-SNSD	noise-test	0.7	51	-	-

Table 11.3: Distributional statistics of speech (top) and noise (bottom) datasets used in `SRB`.

- `treble <gain>`
- `bass <gain>`
- `chorus 0.9 0.9 <delay> 0.4 0.25 2 -t {<delay>+10} 0.3 0.4 2 -s`

Voice Conversion We use use YourTTS [Casanova et al., 2022] from Coqui.ai[1] to synthesize audio from textual transcripts in a given speaker's style. The transcripts from the test clean subset of LibriSpeech are used. The target speakers are drawn from the VCTK corpus [Yamagishi et al., 2019], which contains accented speech from 12 accents. For each transcript a random speaker is chosen to synthesize the audio.

Crosstalk and Music We use crosstalk and music audios from MUSAN [Snyder et al., 2015]. We use `torchaudio.function.add_noise` to add the noise to the speech at a given SNR.

Accents We select a subset of audios from the test set of Common Voice 17. The selected audios satisfied the following criteria: (1) the speaker's accent must be present in the metadata, (2) the accent must not be American, (3) should be clean. The last criterion is satisfied if the DNSMOS [Reddy et al., 2020] score of the recording is at least 3.4. The resulting subset contains 640 recordings. The most popular accent in this set is South Asian (India, Pakistan, Sri Lanka) ($\tilde{2}5\%$), followed by British English ($\tilde{2}5\%$).

Inter-Personal Communications We CHiME-6 [Barker et al., 2017] and AMI [Kraaij et al., 2005] to obtain recordings of people in social scenarios (dinner party and meetings). In both these datasets, the speakers are recorded through lapel microphone and a room microphone resulting in near and far field recordings. We use both types of recordings and show separate results for them. We remove recordings that contain less than three words since they are often fillers.

[1] https://github.com/coqui-ai/TTS

Uttarance Specific Adversarial Attack: The utterance specific adversarial perturbations are computed using the *untargeted* PGD adversarial attack implemented in `robust_speech` package [Olivier and Raj, 2022a]. The attack is computed as follows. First, the maximum possible L2 norm of the noise is determined by solving the equation for SNR for the norm of the noise as follows.

$$\text{SNR} = 20 \log_{10} \left(\frac{||x||_2}{||\delta||_2} \right) \tag{11.1}$$

$$\epsilon_{\text{SNR}} = ||\delta||_2 = 10^{-\frac{\text{SNR}}{20}} ||x||_2, \tag{11.2}$$

where δ is the noise, x is the audio signal and SNR is the upper bound on the SNR in the final signal. Then, we follow the approach of [Madry et al., 2018b] and optimize δ using Projected Gradient Descent (PGD) to maximize the divergence between the true and predicted transcriptions. Formally stated, the attack performs the following optimization:

$$\delta = \max_{\hat{\delta}: \left\|\hat{\delta}\right\|_2 \leq \epsilon_{\text{SNR}}} L_M(x + \hat{\delta}, y), \tag{11.3}$$

where L_M is the loss function used to train the ASR model, M, such as CTCLoss or NLLLoss.

Utterance Agnostic Adversarial Attack: We use the method of [Neekhara et al., 2019], as implemented in `robust_speech` package [Olivier and Raj, 2022a], to compute utterance agnostic adversarial perturbations. The main difference between the universal attack and the PGD attack is that the latter computes a perturbation vector for each input, whereas the former computes a single perturbation that is expected to successfully attack any input to the model.

Formally, given a ASR model, M, and a development speech dataset, $\mathcal{X}^{\text{dev}}$ let $\mathcal{X}^{\text{dev}}_\delta = \{x + \delta | x \in \mathcal{X}^{\text{dev}}\}$ be the same dataset under additive perturbation δ, and let $M(\mathcal{X}^{\text{dev}})$ and $M(\mathcal{X}^{\text{dev}}_\delta$ be the transcripts predicted by M for $\mathcal{X}^{\text{dev}}$ and $\mathcal{X}^{\text{dev}}_\delta$. The utterance agnostic attack uses PGD to optimize δ such that $\|\delta\|_\infty \leq \epsilon$ and the Character-Error Rate (CER) (see § 11.2) between $M(\mathcal{X}^{\text{dev}})$ and $M(\mathcal{X}^{\text{dev}}_\delta)$ is at least t, i.e. $\text{CER}^M(\mathcal{X}^{\text{dev}}_\delta, M(\mathcal{X}^{\text{dev}})) \geq t$ (using the notation from § 11.2). Similar to the utterance-specific attack, the value of ϵ is determined by the maximum allowable SNR using eq.(11.2), except that ℓ_∞ norms are used instead of ℓ_2 norms. The full algorithm is described in Algorith 2.

Once we compute the perturbation we add it to the test audios ($\mathcal{X}^{\text{test}}$) at the specified SNR using `torchaudio.function.add_noise`. For LibriSpeech, we use 500 utterances from test-dev as $\mathcal{X}^{\text{dev}}$ and test-clean as $\mathcal{X}^{\text{dev}}$. For TEDLIUM, we use the full dev and test sets as $\mathcal{X}^{\text{dev}}$ and $\mathcal{X}^{\text{test}}$. For Multi-Lingual LibriSpeech, we use 500 utterances from the dev set in the relevant language as $\mathcal{X}^{\text{dev}}$ and the full test set of the same language as $\mathcal{X}^{\text{test}}$.

11.2 Additional definitions

Word Error Rate: As noted in the main text, we use word error rate (WER) as a basic measure for quantifying performance of the models. Following the common practice from ASR literature, the WER is computed as the word-level edit distance between the reference and the predicted transcripts, normalized by the length of the reference. The edit distance is computed as the total number of word substitutions, deletions, and additions required to transform the reference

Algorithm 2: Utterance Agnostic Attack Algorithm

Require: Speech data $\mathcal{X}^{\text{dev}}$ and , ASR model M and its loss function, L_M (e.g. CTCLoss, NLL-Loss, etc.), allowed SNR s, learning rate, α, max epochs e_{max}, max iterations per sample i_{max}, target attack success rate, t_{sr}, target Character-Error Rate (CER) (see § 11.2), t_{cer}

procedure CER(a, b)
  return EditDistance$(a, b)/len(b)$ ▷ a and b are character sequences
end procedure
procedure SUCCESSRATE($\mathcal{X}$)
  return $\sum_{x \in \mathcal{X}} I[CER(M(x+v), M(x)) > t_{cer}]$
end procedure
procedure SNRTONORM(x, SNR)
  return $10^{-\frac{\text{SNR}}{20}} ||x||_\infty$
end procedure
$\epsilon \leftarrow \sum_{x \in \mathcal{X}^{\text{dev}}} SNRToNorm(x, s)/|\mathcal{X}^{\text{dev}}|$
$v \leftarrow 0$
$e \leftarrow 0$
while $SuccessRate < t_{sr}$ and $e < e_{max}$ **do**
  for $(x, y) \in \mathcal{X}^{\text{dev}}$ **do** ▷ x is the audio, y is the transcript
    $i \leftarrow 0$
    $r \leftarrow 0$
    while $CER(M(x+v+r), M(x)) > t_{cer}$ and $i < i_{max}$ **do**
      $\Delta_r \leftarrow \alpha sign(\nabla_r 0.5 \left\|r\right\|_2 - L_M(x+v+r, y))$
      $r \leftarrow clip_\epsilon\{r - \Delta_r + v\} - v$
      $i \leftarrow i + 1$
    end while
    $v \leftarrow clip_\epsilon\{r + v\}$
  end for
  $e \leftarrow e + 1$
end while

transcript into the predicted transcript. We remove all punctuation from both the reference and predicted transcripts, and convert to lower case before computing WER.

When WER is computed over multiple pairs of predicted and reference transcripts, it is common practice to sum the number of substitutions, deletions, and additions for all the pairs, and divide by the sum of the lengths of the reference transcripts. Formally, this can be written as:

$$WER^M(\mathcal{X}, \mathcal{R}) := 100 \frac{\sum_{x \in \mathcal{X}, r \in \mathcal{R}} ED(M(x), r)}{\sum_{r \in \mathcal{R}} |r|} \%, \tag{11.4}$$

where ED computes the edit distance.

The Character Error Rate (CER) can also be defined similarly, by using the character-level edit distance in the above equation.

For quantifying differences between (binary) genders, we measure the disparity in prediction accuracy across males and females by the Log WER Ratio (LWERR). Formally,

$$LWERR := \log_2 \frac{WER^M(\mathcal{X}_f, \mathcal{R}_f)}{WER^M(\mathcal{X}_m, \mathcal{R}_m)}, \tag{11.5}$$

where the $(\mathcal{X}_f, \mathcal{R}_f)$ and $(\mathcal{X}_m, \mathcal{R}_m)$ represent the subsets of utterances by females and males respectively.

Fairness through robustness: In this work, we use `SRB` to conduct a fairness assessment based on robustness disparities across population subgroups (English vs. Spanish speech; male vs. female speakers). Most of the state of the art quantifies fairness in terms of predictive performance disparities. This occurs in the domain of fairness in ASR systems [Liu et al., 2022, Veliche and Fung, 2023, Koenecke et al., 2020] and similar prevalence is found in other domains [Julia Angwin, Solans Noguero et al., 2023]. However, previous work in the domain of ASR also considered robustness disparities as an alternative fairness notion [Nanda et al., 2021].

Moreover, we argue that considering the dimension of robustness could give better sense of the expected disparities that could be observed when deployed in the wild, in the presence of diverse noisy conditions.

11.3 Models

Table 11.4 provides a summary of the models used in our evaluations. The model names correspond to the names of their pretrained checkpoints in the Huggingface library (https://huggingface.co/models). The abbreviations of these names are in the parentheses after them. Some of the unilingual models are pre-trained on multilingual data but are fine-tuned on only one language and thus can not transcribe any other language. Multilingual models have been pre-trained and fine-tuned on multiple languages so the same DNN can transcribe several languages. The WER of multilingual models is presented as English/Spanish.

11.4 Fine Grained Analyses

The following figures present fine-grained analyses of robustness. These figures may be referenced by the main text but were not included in the main body in the interest of space. Figure 11.1

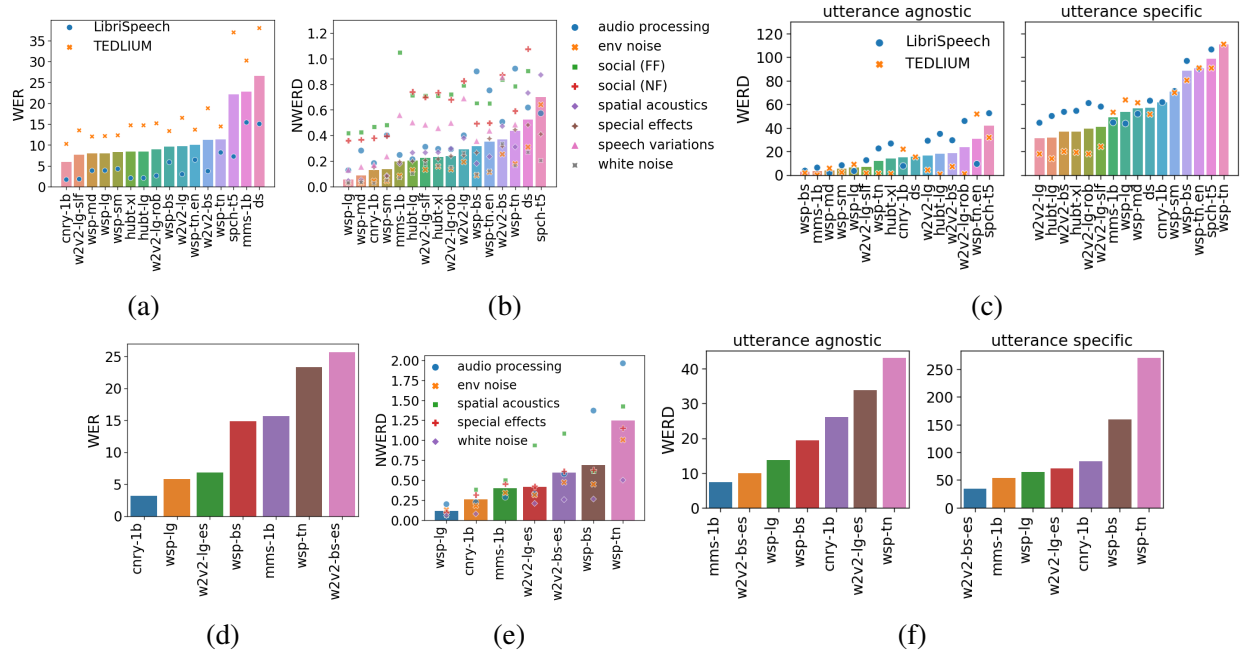


Figure 11.1: The accuracy and robustness of English (top) and Spanish (bottom) ASR models on clean and perturbed data. Accuracy is measured by WER of the models on clean speech (a & d). Robustness is measured by the NWERD on non-adversarially perturbed speech (b & e) and WERD on adversarially perturbed speech (c & f). The markers indicate the dataset in (a & c), and the perturbation category in (b & e). The x axes are in ascending order of the values on the y axes.

provides an overview of the accuracy and robustness of the various models. Figures 11.2, and 11.3 present the breakdown by perturbation of the robustness of models on English and Spanish, respectively. Figure 11.4 presents a breakdown of robustness by severity.

11.5 Compute Resources

The experiments were performed on the Bridges-2 cluster at the Pittsburgh Supercomputing Center. This cluster contains 200 32G and 16G Nvidia V-100, which were used for these experiments.

11.6 Dataset Licenses

The licenses of each of the considered datasets are described in Table 11.5.

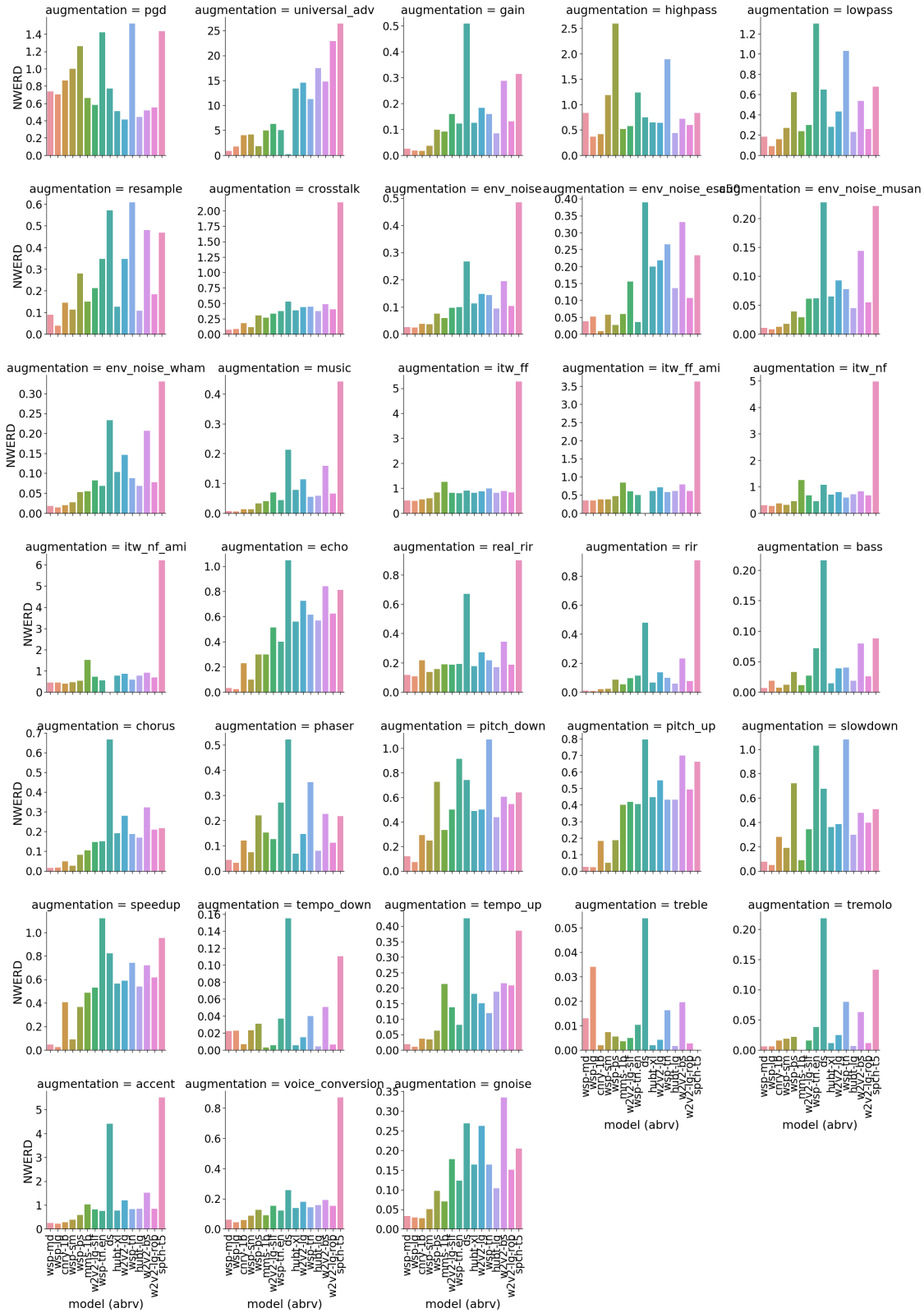


Figure 11.2: NWERD of English models on different augmentations, averaged over all severities.

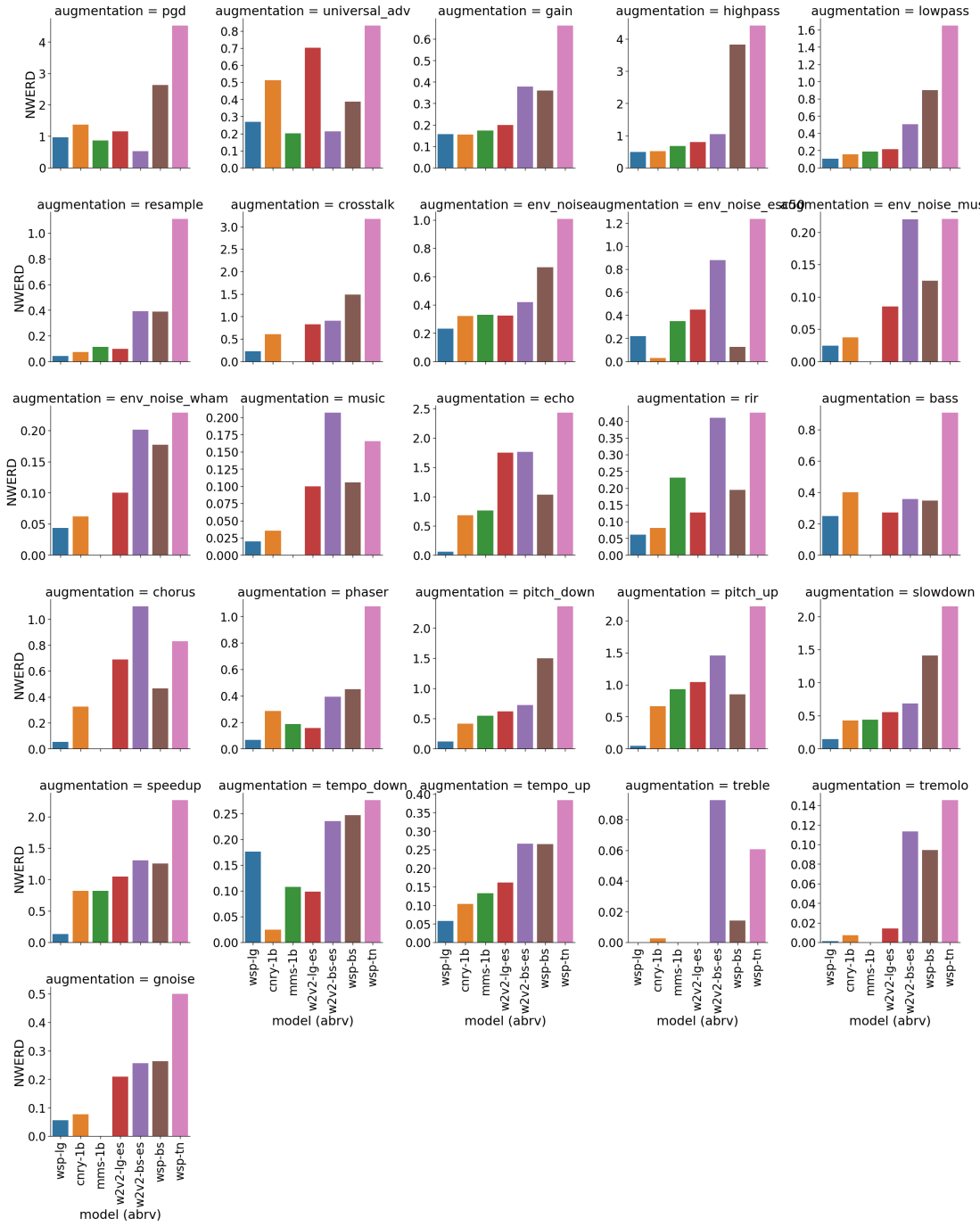


Figure 11.3: NWERD of Spanish models on different augmentations, averaged over all severities.

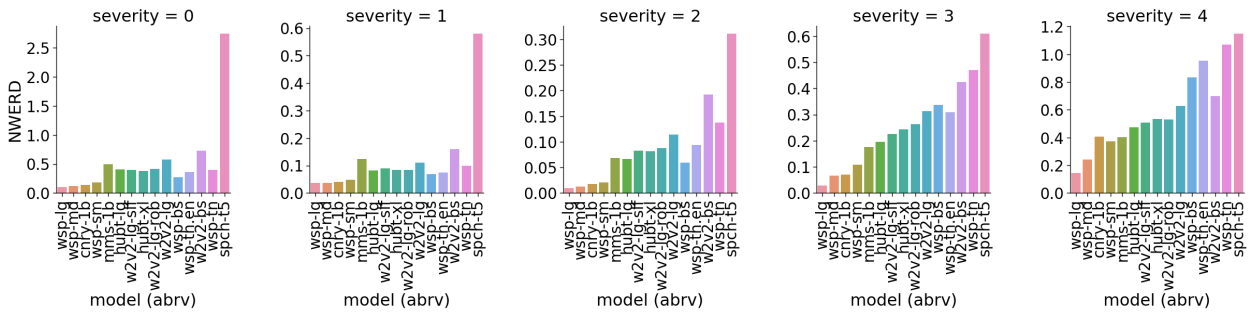
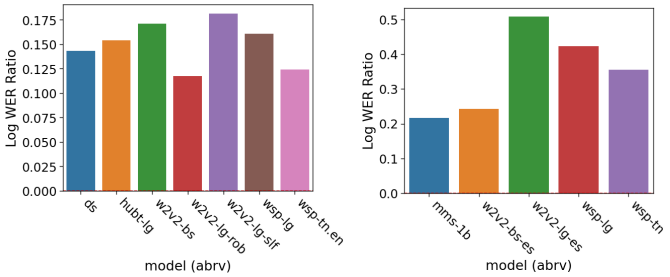


Figure 11.4: NWERD on English data as the severity of the augmentation is increased.



(a) (b)

Figure 11.5: Log WER Ratio between male and female speakers from Librispeech (English) (a) and Spanish Multilingual Librispeech (b).

lang	dataset	category model (abrv)	clean WER	accent	audio proc	noise (env)	noise (white)	sFX NWERD	social (FF)	social (NF)	spatial	synth speech	AVG	Adv (UA) WERD	Adv (US)	AVG
du	MLS-DU	cnry-1b	4.0	-	15.5	5.3	6.0	12.4	-	-	12.2	-	10.3	-	-	-
		wsp-lg	7.3	-	14.3	2.6	6.5	4.1	-	-	5.3	-	6.6	-	-	-
		mms-1b	14.6	-	21.6	8.1	12.0	14.3	-	-	18.9	-	15.0	-	-	-
		wsp-bs	20.4	-	93.8	14.0	26.3	34.5	-	-	24.7	-	38.6	-	-	-
		prkt-rnnt-1.1b	1.6	-	5.1	2.6	1.8	3.7	-	-	8.0	3.0	4.0	-	-	-
		cnry-1b	1.7	-	12.3	3.0	2.7	12.7	-	-	9.3	4.2	7.4	7.7	61.7	34.7
		prkt-rnnt-0.6b	1.8	-	11.2	3.4	3.0	6.0	-	-	9.4	3.8	6.1	-	-	-
		w2v2-lg-slf	1.8	-	16.9	9.2	17.1	12.4	-	-	14.4	5.5	12.6	12.5	58.0	35.2
		prkt-ctc-0.6b	2.0	-	10.9	3.5	3.4	7.3	-	-	10.4	4.7	6.7	-	-	-
		prkt-ctc-1.1b	2.0	-	6.2	3.0	3.3	6.4	-	-	9.1	4.4	5.4	-	-	-
		hubt-xl	2.1	-	18.2	11.1	16.3	13.3	-	-	14.7	5.5	13.2	26.7	54.4	40.5
		hubt-lg	2.1	-	11.6	9.0	9.1	12.3	-	-	15.0	6.5	10.6	34.9	50.0	42.5
		sb-cnfmr	2.6	-	9.4	11.0	15.2	11.3	-	-	19.0	7.4	12.2	-	-	-
		w2v2-lg-rob	2.6	-	18.1	10.2	15.1	15.4	-	-	16.9	5.9	13.6	45.8	61.0	53.4
		sb-cnfmr-rnnt	2.7	-	15.7	9.8	17.2	9.6	-	-	20.4	8.4	13.5	-	-	-
		w2v2-lg	3.0	-	24.7	14.4	28.2	15.8	-	-	21.8	8.6	18.9	29.0	44.3	36.6
	LibriSpeech	w2v2-bs	3.7	-	30.6	18.9	36.4	20.3	-	-	26.9	9.6	23.8	29.5	53.6	41.5
		wsp-md	3.9	-	19.1	2.4	3.3	2.1	-	-	3.8	5.3	6.0	1.7	51.9	26.8
		wsp-lg	3.9	-	6.9	2.0	3.5	1.6	-	-	3.0	4.0	3.5	3.4	43.6	23.5
		wsp-sm.en	4.0	-	34.9	6.9	-	6.0	-	-	3.9	6.2	11.6	-	-	-
		wsp-md.en	4.1	-	16.0	4.9	-	2.8	-	-	0.9	4.8	5.9	-	-	-
		wsp-sm	4.3	-	27.9	4.0	5.4	4.5	-	-	6.5	7.0	9.2	8.3	71.1	39.7
		wsp-bs.en	5.1	-	48.0	12.4	-	17.9	-	-	11.2	8.1	19.5	-	-	-
		wsp-bs	5.9	-	63.6	6.6	11.0	14.8	-	-	14.5	8.5	19.8	3.6	96.7	50.2
		wsp-tn.en	6.4	-	57.2	8.6	14.0	25.8	-	-	17.1	9.2	22.0	9.4	90.2	49.8
		spch-t5	7.2	-	32.6	49.8	26.2	27.3	-	-	67.9	64.3	44.7	52.4	106.4	79.4
		sbcrdnn	7.2	-	-	11.5	33.9	-	-	-	-	-	22.7	-	-	-
		wsp-tn	8.2	-	68.4	13.8	17.4	26.1	-	-	21.8	9.9	26.2	22.5	-	22.5
		ds	15.1	-	37.9	26.2	28.9	32.0	-	-	46.2	18.0	31.5	-	62.9	62.9
		mms-1b	15.4	-	16.3	5.5	7.0	11.1	-	-	11.9	8.9	10.1	6.2	44.6	25.4
		cnry-1b	10.2	-	15.3	3.1	2.9	10.6	-	-	11.7	1.5	7.5	21.9	-	21.9
		wsp-md	12.0	-	22.9	1.7	3.5	3.1	-	-	4.3	0.8	6.1	5.7	61.3	33.5
		wsp-lg	12.1	-	12.5	2.5	2.6	2.3	-	-	4.0	0.4	4.0	9.0	63.6	36.3
		wsp-sm	12.3	-	32.0	2.3	4.9	6.1	-	-	5.6	1.6	8.8	2.2	69.8	36.0
		wsp-bs	13.3	-	67.6	5.8	8.8	19.2	-	-	8.7	4.1	19.0	1.7	80.3	41.0
		w2v2-lg-slf	13.5	-	27.6	9.5	19.3	17.2	-	-	18.8	9.9	17.0	1.5	24.0	12.8
		wsp-tn.en	13.7	-	50.9	7.2	10.9	29.3	-	-	12.9	2.8	19.0	51.7	90.7	71.2
	TEDLIUM	wsp-tn	14.4	-	59.7	11.5	15.8	29.9	-	-	16.2	4.3	22.9	1.3	110.9	56.1
		hubt-lg	14.7	-	19.9	9.2	12.1	16.2	-	-	17.5	9.1	14.0	0.7	13.9	7.3
		hubt-xl	14.7	-	24.6	11.1	17.3	16.9	-	-	18.2	8.2	16.0	1.2	19.1	10.2
		w2v2-lg-rob	15.2	-	24.4	8.7	15.7	18.7	-	-	19.4	9.2	16.0	1.0	17.9	9.4

		w2v2-lg	16.5	-	31.2	12.9	24.7	18.8	-	-	25.5	9.2	20.4	4.2	17.9	11.0
		w2v2-bs	18.8	-	37.0	16.5	30.0	23.3	-	-	32.6	9.5	24.8	7.1	19.9	13.5
		mms-1b	30.2	-	19.0	6.2	7.3	12.1	-	-	11.5	0.0	9.4	0.0	53.1	26.6
		spch-t5	37.0	-	48.8	31.8	15.2	27.5	-	-	48.1	21.7	32.2	31.8	90.6	61.2
		ds	38.0	-	40.1	16.1	24.3	30.0	-	-	48.7	7.3	27.8	15.3	51.5	33.4
		cnry-1b	-	-	-	-	-	-	31.7	13.9	-	-	22.8	-	-	-
		hubt-lg	-	-	-	-	-	-	51.0	27.7	-	-	39.4	-	-	-
		hubt-xl	-	-	-	-	-	-	51.1	27.6	-	-	39.4	-	-	-
		mms-1b	-	-	-	-	-	-	71.2	55.2	-	-	63.2	-	-	-
		prkt-ctc-0.6b	-	-	-	-	-	-	34.5	14.5	-	-	24.5	-	-	-
		prkt-ctc-1.1b	-	-	-	-	-	-	31.6	13.6	-	-	22.6	-	-	-
		prkt-rnnt-0.6b	-	-	-	-	-	-	37.0	16.7	-	-	26.9	-	-	-
		prkt-rnnt-1.1b	-	-	-	-	-	-	35.9	17.0	-	-	26.4	-	-	-
		sb-cnfmr	-	-	-	-	-	-	63.0	35.4	-	-	49.2	-	-	-
		sb-cnfmr-rnnt	-	-	-	-	-	-	58.6	31.6	-	-	45.1	-	-	-
	ami	sbcrdnn	-	-	-	-	-	-	91.2	-	-	-	91.2	-	-	-
		spch-t5	-	-	-	-	-	-	304.3	221.9	-	-	263.1	-	-	-
		w2v2-bs	-	-	-	-	-	-	66.0	32.9	-	-	49.4	-	-	-
		w2v2-lg	-	-	-	-	-	-	59.6	30.8	-	-	45.2	-	-	-
		w2v2-lg-rob	-	-	-	-	-	-	51.1	24.7	-	-	37.9	-	-	-
		w2v2-lg-slf	-	-	-	-	-	-	50.6	26.2	-	-	38.4	-	-	-
		wsp-bs	-	-	-	-	-	-	39.5	19.1	-	-	29.3	-	-	-
		wsp-lg	-	-	-	-	-	-	29.3	15.9	-	-	22.6	-	-	-
		wsp-md	-	-	-	-	-	-	29.2	15.8	-	-	22.5	-	-	-
		wsp-sm	-	-	-	-	-	-	31.4	17.0	-	-	24.2	-	-	-
		wsp-tn	-	-	-	-	-	-	48.9	21.2	-	-	35.0	-	-	-
		wsp-tn.en	-	-	-	-	-	-	41.7	19.5	-	-	30.6	-	-	-
		cnry-1b	-	-	-	-	-	-	55.6	28.8	-	-	42.2	-	-	-
		ds	-	-	-	-	-	-	90.6	84.0	-	-	87.3	-	-	-
		hubt-lg	-	-	-	-	-	-	82.1	55.2	-	-	68.7	-	-	-
		hubt-xl	-	-	-	-	-	-	81.1	54.6	-	-	67.8	-	-	-
		mms-1b	-	-	-	-	-	-	126.7	98.8	-	-	112.7	-	-	-
		prkt-ctc-0.6b	-	-	-	-	-	-	56.6	27.4	-	-	42.0	-	-	-
		prkt-ctc-1.1b	-	-	-	-	-	-	53.4	26.7	-	-	40.0	-	-	-
		prkt-rnnt-0.6b	-	-	-	-	-	-	60.5	28.3	-	-	44.4	-	-	-
		prkt-rnnt-1.1b	-	-	-	-	-	-	55.9	27.3	-	-	41.6	-	-	-
		sb-cnfmr	-	-	-	-	-	-	92.5	74.2	-	-	83.3	-	-	-
		sb-cnfmr-rnnt	-	-	-	-	-	-	84.3	60.5	-	-	72.4	-	-	-
	chime	sbcrdnn	-	-	-	-	-	-	92.9	87.3	-	-	90.1	-	-	-
		spch-t5	-	-	-	-	-	-	527.7	388.7	-	-	458.2	-	-	-
		w2v2-bs	-	-	-	-	-	-	88.7	64.6	-	-	76.7	-	-	-
		w2v2-lg	-	-	-	-	-	-	86.9	61.7	-	-	74.3	-	-	-
		w2v2-lg-rob	-	-	-	-	-	-	83.4	52.3	-	-	67.8	-	-	-
		w2v2-lg-slf	-	-	-	-	-	-	81.6	52.0	-	-	66.8	-	-	-
		wsp-bs	-	-	-	-	-	-	83.4	35.5	-	-	59.4	-	-	-
		wsp-lg	-	-	-	-	-	-	48.6	21.6	-	-	35.1	-	-	-
		wsp-md	-	-	-	-	-	-	50.6	22.8	-	-	36.7	-	-	-
		wsp-sm	-	-	-	-	-	-	59.2	24.9	-	-	42.1	-	-	-
		wsp-tn	-	-	-	-	-	-	98.6	46.3	-	-	72.4	-	-	-
		wsp-tn.en	-	-	-	-	-	-	80.4	35.2	-	-	57.8	-	-	-
		cnry-1b	-	4.6	-	-	-	-	-	-	-	-	4.6	-	-	-
		ds	-	93.4	-	-	-	-	-	-	-	-	93.4	-	-	-
		hubt-lg	-	14.1	-	-	-	-	-	-	-	-	14.1	-	-	-
		hubt-xl	-	12.9	-	-	-	-	-	-	-	-	12.9	-	-	-
		mms-1b	-	18.1	-	-	-	-	-	-	-	-	18.1	-	-	-
		prkt-ctc-0.6b	-	5.8	-	-	-	-	-	-	-	-	5.8	-	-	-
		prkt-ctc-1.1b	-	5.6	-	-	-	-	-	-	-	-	5.6	-	-	-
		prkt-rnnt-0.6b	-	5.1	-	-	-	-	-	-	-	-	5.1	-	-	-
		prkt-rnnt-1.1b	-	3.8	-	-	-	-	-	-	-	-	3.8	-	-	-
		sb-cnfmr	-	20.3	-	-	-	-	-	-	-	-	20.3	-	-	-
		sb-cnfmr-rnnt	-	19.1	-	-	-	-	-	-	-	-	19.1	-	-	-
	CV	sbcrdnn	-	44.8	-	-	-	-	-	-	-	-	44.8	-	-	-
		spch-t5	-	89.7	-	-	-	-	-	-	-	-	89.7	-	-	-
		w2v2-bs	-	25.4	-	-	-	-	-	-	-	-	25.4	-	-	-
		w2v2-lg	-	20.0	-	-	-	-	-	-	-	-	20.0	-	-	-
		w2v2-lg-rob	-	13.9	-	-	-	-	-	-	-	-	13.9	-	-	-
		w2v2-lg-slf	-	13.5	-	-	-	-	-	-	-	-	13.5	-	-	-
		wsp-bs	-	10.0	-	-	-	-	-	-	-	-	10.0	-	-	-
		wsp-lg	-	3.7	-	-	-	-	-	-	-	-	3.7	-	-	-
		wsp-md	-	4.1	-	-	-	-	-	-	-	-	4.1	-	-	-
		wsp-sm	-	6.5	-	-	-	-	-	-	-	-	6.5	-	-	-
		wsp-tn	-	13.9	-	-	-	-	-	-	-	-	13.9	-	-	-
		wsp-tn.en	-	12.7	-	-	-	-	-	-	-	-	12.7	-	-	-
		cnry-1b	3.2	-	16.4	9.6	8.2	16.0	-	-	10.9	-	12.2	26.1	84.3	55.2
		wsp-lg	5.8	-	14.1	7.4	5.9	4.0	-	-	2.0	-	6.7	13.7	65.0	39.4
		w2v2-lg-es	6.8	-	21.2	14.7	22.1	19.0	-	-	26.3	-	20.6	33.9	71.0	52.4
	MLS-ES	wsp-bs	14.8	-	90.9	22.7	27.3	32.5	-	-	18.0	-	38.3	19.5	159.5	89.5
		mms-1b	15.7	-	18.5	19.7	37.1	20.0	-	-	14.9	-	22.0	7.4	53.8	30.6
		wsp-tn	23.3	-	124.3	45.0	52.3	57.9	-	-	41.8	-	64.3	43.1	269.9	156.5
		w2v2-bs-es	25.7	-	32.4	20.4	24.2	24.1	-	-	32.4	-	26.7	10.0	33.8	21.9
es		cnry-1b	-	205.0	-	-	-	-	-	-	-	-	205.0	-	-	-
		mms-1b	-	7.7	-	-	-	-	-	-	-	-	7.7	-	-	-
		w2v2-bs-es	-	13.3	-	-	-	-	-	-	-	-	13.3	-	-	-
	CV:es	w2v2-lg-es	-	9.3	-	-	-	-	-	-	-	-	9.3	-	-	-
		wsp-bs	-	18.2	-	-	-	-	-	-	-	-	18.2	-	-	-
		wsp-lg	-	2.0	-	-	-	-	-	-	-	-	2.0	-	-	-
		wsp-tn	-	30.6	-	-	-	-	-	-	-	-	30.6	-	-	-
		cnry-1b	6.1	-	15.2	5.2	7.3	13.0	-	-	10.0	-	10.1	-	-	-
		wsp-lg	7.7	-	15.5	3.5	8.3	5.6	-	-	5.7	-	7.7	-	-	-
fr	MLS-FR	mms-1b	23.6	-	21.0	6.9	12.5	12.9	-	-	15.6	-	13.8	-	-	-
		wsp-bs	26.0	-	124.8	16.7	44.4	47.9	-	-	25.9	-	51.9	-	-	-

Table 11.6: Accuracy and robustness of ASR models on all datasets

Lang	Model	Abrv.	Params (M)	Data (hrs)	WER
EN	canary-1b [NVIDIA]	cnry-1b	1,000.0	85,000	6.0
	parakeet-ctc-0.6b [NVIDIA]	prkt-ctc-0.6b	600.0	64,000.0	6.1
	parakeet-ctc-1.1b [NVIDIA]	prkt-ctc-1.1b	1,100.0	64,000.0	6.0
	parakeet-rnnt-0.6b [NVIDIA]	prkt-rnnt-0.6b	600.0	64,000.0	6.0
	parakeet-rnnt-1.1b [NVIDIA]	prkt-rnnt-1.1b	1,100.0	64,000.0	5.9
	deepspeech[Amodei et al., 2016]	ds	86.0	960	26.5
	hubert-large-ls960-ft[Hsu et al., 2021a]	hubt-lg	317.0	60,000	8.4
	hubert-xlarge-ls960-ft[Hsu et al., 2021a]	hubt-xl	964.0	60,000	8.4
	mms-1b-fl102 [Pratap et al., 2023]	mms-1b	964.6	55,000	22.8
	speecht5 asr[Ao et al., 2022]	spch-t5	154.6	960	22.1
	wav2vec2-base-960h [Baevski et al., 2020]	w2v2-bs	95.0	960	11.3
	wav2vec2-large-960h [Baevski et al., 2020]	w2v2-lg	317.0	960	9.7
	wav2vec2-large-960h-lv60-self [Xu et al., 2021]	w2v2-lg-slf	317.0	60,000	7.7
	wav2vec2-large-robust-ft-libri-960h [Hsu et al., 2021b]	w2v2-lg-rob	317.0	63,000	8.9
	whisper-base [Radford et al., 2023]	wsp-bs	74.0	680,000	9.6
	whisper-base.en [Radford et al., 2023]	wsp-bs.en	74.0	563,000	5.1
	whisper-large-v2 [Radford et al., 2023]	wsp-lg	1,550.0	680,000	8.0
	whisper-medium [Radford et al., 2023]	wsp-md	769.0	680,000	7.9
	whisper-medium.en [Radford et al., 2023]	wsp-md.en	769.0	563,000	4.1
	whisper-small [Radford et al., 2023]	wsp-sm	244.0	680,000	8.3
	whisper-small.en [Radford et al., 2023]	wsp-sm.en	244.0	563,000	4.0
	whisper-tiny [Radford et al., 2023]	wsp-tn	39.0	680,000	11.3
	whisper-tiny.en [Radford et al., 2023]	wsp-tn.en	39.0	563,000	10.1
ES	canary-1b [NVIDIA]	cnry-1b	1,000.0	85,000	3.2
	mms-1b-fl102 [Pratap et al., 2023]	mms-1b	964.6	55,000	15.7
	wav2vec2-base-10k-voxpopuli-ft-es [Wang et al., 2021b]	w2v2-bs-es	94.4	10,116	25.7
	wav2vec2-large-xlsr-53-spanish [Conneau et al., 2020]	w2v2-lg-es	315.4	54,350	6.8
	whisper-base [Radford et al., 2023]	wsp-bs	74.0	680,000	14.8
	whisper-large-v2 [Radford et al., 2023]	wsp-lg	1,550.0	680,000	5.8
	whisper-tiny [Radford et al., 2023]	wsp-tn	39.0	680,000	23.3

Table 11.4: Models used in our evaluations.

Dataset	License
LibriSpeech	CC-BY-4.0
Multilingual Librispeech	CC BY 4.0
TEDLIUM	CC-BY-NC-ND 3.0
AMI	CC-BY-4.0
Common Voice	CC0-1.0
CHiME	CC BY-SA 4.0

Table 11.5: Licenses of each of the considered datasets in `SRB`