WaveGuard: Understanding and Mitigating Audio Adversarial Examples

*Shehzeen Hussain, *Paarth Neekhara, Shlomo Dubnov, Julian McAuley, Farinaz Koushanfar
University of California San Diego
{ssh028,pneekhar}@ucsd.edu
* Equal contribution

Abstract

There has been a recent surge in adversarial attacks on deep learning based automatic speech recognition (ASR) systems. These attacks pose new challenges to deep learning security and have raised significant concerns in deploying ASR systems in safety-critical applications. In this work, we introduce WaveGuard: a framework for detecting adversarial inputs that are crafted to attack ASR systems. Our framework incorporates audio transformation functions and analyses the ASR transcriptions of the original and transformed audio to detect adversarial inputs.[1] We demonstrate that our defense framework is able to reliably detect adversarial examples constructed by four recent audio adversarial attacks, with a variety of audio transformation functions. With careful regard for best practices in defense evaluations, we analyze our proposed defense and its strength to withstand adaptive and robust attacks in the audio domain. We empirically demonstrate that audio transformations that recover audio from perceptually informed representations can lead to a strong defense that is robust against an adaptive adversary even in a complete white-box setting. Furthermore, WaveGuard can be used out-of-the box and integrated directly with any ASR model to efficiently detect audio adversarial examples, without the need for model retraining.

1 Introduction

Speech serves as a powerful communication interface between humans and machine learning agents. Speech interfaces enable hands-free operation and can assist users who are visually or physically impaired. Research into machine recognition of speech is driven by the prospect of offering services where humans interact naturally with machines. To this end, automatic speech recognition (ASR) systems seek to accurately convert a speech signal into a transcription of the spoken words, irrespective of a speaker's accent, or the acoustic environment in which the speaker is located [1]. With the advent of deep learning, state-of-the-art speech recognition systems [2–4] are based on Deep Neural Networks (DNNs) and are widely used in personal assistants and home electronic devices (e.g. Apple Siri, Google Assistant).

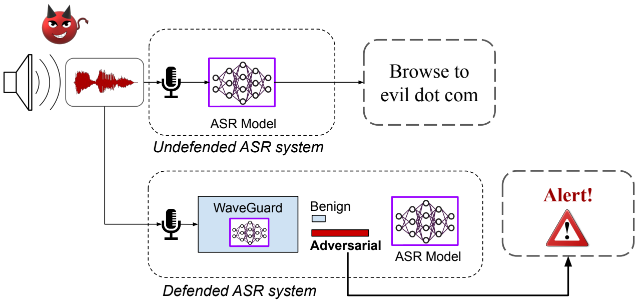


Figure 1: Depiction of an undefended ASR system and an ASR system defended by WaveGuard in the presence of a malicious adversary. The ASR system defended by WaveGuard detects the adversarial input and alerts the user.

The popularity of ASR systems has brought new security concerns. Several studies have demonstrated that DNNs are vulnerable to adversarial examples [5–8]. While previously limited to the image domain, recent attacks on ASR systems [9–17], have demonstrated that adversarial examples also exist in the audio domain. An audio adversarial example can cause the original audio signal to be transcribed to a target phrase desired by the adversary or can cause significant transcription error by the victim ASR model.

Due to the existence of these vulnerabilities, there is a crucial need for defensive methods that can be employed to thwart audio adversarial attacks. In the image domain, several works have proposed input transformation based defenses [18–22] to recover benign images from adversarially modified images. Such inference-time adversarial defenses use image transformations like feature squeezing, JPEG compression, quantization, randomized smoothing (etc.) to render adversarial examples ineffective. While such defenses are effective in guarding against non-adaptive adversaries, they can

[1] Audio Examples: https://waveguard.herokuapp.com

be bypassed in an adaptive attack scenario where the attacker has partial or complete knowledge about the defense.

Another line of defense in the image domain is based on training more robust neural networks using adversarial training or by introducing randomization in network layers and parameters. Such defenses are comparatively more robust under adaptive attack scenarios, however they are significantly more expensive to train as compared to input transformation based defenses that can be employed directly at the model inference stage. Although input transformation based defenses are shown to be broken for image classifiers, the same conclusion cannot be drawn for ASR systems without careful evaluation. This is because an ASR system is a more complicated architecture as compared to an image classification model and involves several individual components: an acoustic feature extraction pipeline, a neural sequence model for processing the time-series data and a language head for predicting the language tokens. This pipeline makes it challenging to craft robust adversarial examples for ASR systems that can reliably transcribe to a target phrase even when the input is transformed and reconstructed from some perceptually informed representation.

WaveGuard: In this work, we study the effectiveness of audio transformation based defenses for detecting adversarial examples for speech recognition systems. We first design a general framework for employing audio transformation functions as an adversarial defense for ASR systems. Our framework transforms the given audio input x using an input transformation function g and analyzes the ASR transcriptions for the input x and $g(x)$. The underlying idea for our defense is that model predictions for adversarial examples are unstable while those for benign examples are robust to small changes in the input. Therefore, our framework labels an input as adversarial if there is a significant difference between the transcriptions of x and $g(x)$.

We first study five different audio transformations under different compression levels against non-adaptive adversaries. We find that at optimal compression levels, most input transformations can reliably discriminate between adversarial and benign examples for both targeted and untargeted adversarial attacks on ASR systems. Furthermore, we achieve higher detection accuracy in comparison to prior work [23, 24] in adversarial audio detection. However, this evaluation does not provide security guarantees against a future adaptive adversary who has knowledge of our defense framework. To evaluate the robustness of our defense against an adaptive adversary, we propose a strong white-box adaptive attack against our proposed defense framework. Interestingly, we find that some input transformation functions are robust to adaptive attack even when the attacker has complete knowledge of the defense. Particularly, the transformations that recover audio from perceptually informed representations of speech prove to be more effective against adaptive-attacks than naive audio compression and filtering techniques.

Summary of Contributions:

- We develop a formal defense framework (Section 3) for detecting audio adversarial examples against ASR systems. Our framework uses input transformation functions and analyses the transcriptions of original and transformed audio to label the input as adversarial or benign.
- We evaluate different transformation functions for detecting recently proposed and highly successful targeted [11, 14] and untargeted [15] attacks on ASR systems. We study the trade-off between the hyperparameters of different transformations and the detector performance and find an optimal range of hyperparameters for which the given transformation can reliably detect adversarial examples (Section 6).
- We demonstrate the robustness of our defense framework against an adaptive adversary who has complete knowledge of our defense and intends to bypass it. We find that certain input transformation functions that reduce audio to a perceptually informed representation cannot be easily bypassed under different allowed magnitudes of perturbations. Particularly, we find that Linear Predictive Coding (LPC) and Mel spectrogram extraction-inversion are more robust to adaptive attacks as compared to other transformation functions studied in our work (Section 7).
- We investigate transformation functions for the goal of recovering the original transcriptions from an adversarial signal. We find that for certain attacks and transformation functions, we can recover the original transcript with a low Character Error Rate. (Section 6.2)

2 Background and Related Work

2.1 Adversarial Attacks in the Audio Domain:

Adversarial attacks on ASR systems have primarily focused on *targeted attacks* to embed carefully crafted perturbations into speech signals, such that the victim model transcribes the input audio into a specific malicious phrase, as desired by the adversary [9, 11, 12, 25, 26]. Such attacks can for example cause a digital assistant to incorrectly recognize commands it is given, thereby compromising the security of the device. Prior works [12, 26] demonstrate successful attack algorithms targeting traditional speech recognition models based on HMMs and GMMs [27–32]. For example, in Hidden Voice Commands [12], the attacker uses inverse feature extraction to generate obfuscated audio that can be played over-the-air to attack ASR systems. However, obfuscated samples sound like random noise rather than normal human perceptible speech and therefore come at the cost of being fairly perceptible to human listeners.

In more recent work [11] involving neural network based ASR systems, Carlini *et al.* propose an end-to-end white-box attack technique to craft adversarial examples, which transcribe to a target phrase. Similar to work in images, they propose a gradient-based optimization method that replaces the cross-entropy loss function used for classification, with a Connectionist Temporal Classification (CTC) loss [33] which is optimized for time-sequences. The CTC-loss between the target phrase and the network's output is backpropagated through the victim neural network and the Mel Frequency Cepstral Coefficient (MFCC) computation, to update the additive adversarial perturbation. The authors in this work demonstrate 100% attack success rate on the Mozilla DeepSpeech [4] ASR model. The adversarial samples generated by this work are quasi-perceptible, motivating a separate work [10] to minimize the perceptibility of the adversarial perturbations using psychoacoustic hiding. Further addressing the imperceptibility of audio attacks, Qin *et al.* [14] develop effectively imperceptible audio adversarial examples by leveraging the psychoacoustic principle of auditory masking. In their work [14], the imperceptibility of adversarial audio is verified through a human study, while retaining 100% targeted attack success rate on the Google Lingvo [3] ASR model.

Targeted attacks, such as those described above, cannot be performed in real-time since it requires the adversary to solve a data-dependent optimization problem for each datapoint they wish to mis-transcribe. To perform attacks in real-time, the authors of [15] designed an algorithm to find a single quasi-imperceptible universal perturbation, which when added to any arbitrary speech signal, causes mis-transcription by the victim speech recognition model. The proposed algorithm iterates over the training dataset to build a universal perturbation vector, that can be added to any speech waveform to cause an error in transcription by a speech recognition model with high probability. This work also demonstrates transferability of adversarial audio samples across two different ASR systems (based on DeepSpeech and Wavenet), demonstrating that such audio attacks can be performed in real-time even when the attacker does not have knowledge of the ASR model parameters.

Physical attacks. Adversarial attacks to ASR Systems have also been demonstrated to be a real-world threat. In particular, recently developed attack algorithms have shown success in attacking physical intelligent voice control (IVC) devices, when playing the generated adversarial examples over-the-air. The recently developed *Devil's Whisper* [17] demonstrated that adversarial commands embedded in music samples and played over-the-air using speakers, are able to attack popular IVC devices such as Google Home, Google Assistant, Microsoft Cortana and Amazon Alexa with 98% of target commands being successful. They utilize a surrogate model approach to generate transferable adversarial examples that can attack a number of unseen target devices. However, as noted by the authors, physical attacks are very sensitive to various environmental factors, such as the volume when playing adversarial examples, the distance between the speaker and the victim IVC device, as well as the brand of speakers, that can render the attack unsuccessful. Qin *et al.* [14] designed robust, physical-world, over-the-air audio adversarial examples by constructing perturbations, which remain effective in attacking the Google Lingvo ASR model [3] even after applying environmental distortions. Such robust adversarial examples are crafted by incorporating the noise simulation during the training process of the perturbation. In our work, we evaluate our defense against the robust attack proposed in [14] on the Google Lingvo ASR model. We find that while such examples are more robust to small input changes as compared to previously proposed targeted attacks [11], they can still be easily distinguished from benign audio samples using our defense framework.

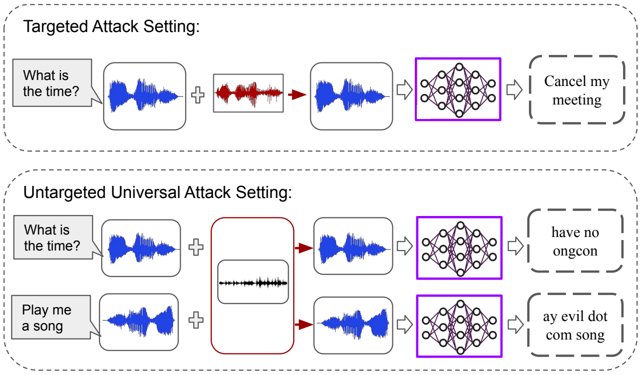


Figure 2: **Top:** In the targeted attack setting, the adversary solves a data-dependent optimization problem to find an additive perturbation, such that a victim ASR model transcribes the adversarial input audio to a target phrase as desired by the adversary. **Bottom:** In the untargeted universal attack setting, the adversary computes a single universal perturbation which when added to any arbitrary audio signal, will most likely cause an error in transcription by a victim ASR system. In untargeted attacks, the transcription of adversarial audio may not be a specific malicious phrase.

2.2 Principles of Defense and Adaptive Attacks in the Image Domain

To strengthen the reliability of deep learning models in the image domain, a significant amount of prior work has proposed defenses to adversarial attacks [18–20, 22, 34, 35]. However, most of these defenses were only evaluated against non-adaptive attacks or using a "zero-knowledge" threat model, where the attacker has no knowledge of the defense existing in the system. Such defenses offer bare-minimum security and in no way guarantee that they can be secure against future attacks [36, 37]. Accurately evaluating the robustness of defenses is a challenging but important task, particularly because of the presence of adaptive adversaries [6,37–39]. An adaptive adversary is one that has partial or complete knowledge of the defense mechanism in place and therefore adapts their attack to what the defender has designed [37, 38, 40].

Many prior works on defenses are variants of the same idea: pre-process inputs using a transform, e.g. randomized cropping, rotation, JPEG compression, randomized smoothing, auto-encoder transformation, that can remove the adversarial perturbation from the input. However, such defenses are shown to be vulnerable to attack algorithms that are partially or completely aware of the defense mechanism [6, 41]. In [6], the authors show that the input-transformation function can be substituted with a differentiable approximation in the backward pass in-order to craft adversarial examples that are robust under the given input-transform. In [41], the authors craft adversarial examples that are robust over a given distribution of transformation functions, which guarantees robustness over more than one type of transform.

Solely analyzing a defense against a non-adaptive adversary gives us a false sense of security. Therefore, the authors of [37] provided several guidelines to ensure completeness in the evaluation of defenses to adversarial attacks. The authors recommend using a threat model with an "infinitely thorough" adaptive adversary, who is capable of developing new optimal attacks against the proposed defense. They recommend applying a diverse set of attacks to any proposed defense, with the same mindset of a future adversary. However, such defense guidelines have not been applied to the audio domain and many of the proposed ASR defenses have not carried out thorough evaluations against adaptive adversaries. In our work, we follow these guidelines and evaluate our ASR defense against the strongest non-adaptive and adaptive adversaries.

2.3 Defenses in the Audio Domain

In comparison to the image domain, only a handful of studies have proposed defenses to adversarial attacks in the audio domain. Prior work on defenses for speech recognition models have focused on both audio pre-processing techniques [23,42] and utilizing temporal dependency in speech signals [24] to detect adversarial examples.

Yang *et al.* in [24] proposed a defense framework against three attack methods targeting state-of-the-art ASR models such as Kaldi and DeepSpeech. The proposed defense framework checks if the transcription of the first k-sized portion of the audio waveform (t_1) is similar to the first k-sized transcription of the complete audio waveform (t_2). A sample is identified as adversarial when the two transcriptions are dissimilar, i.e., the Character Error Rate (CER) or Word Error Rate (WER) between t_1 and t_2 is higher than a predefined threshold. The authors further study the effectiveness of their defense in an adaptive attack scenario, where the attacker has partial knowledge of the defense framework. In their strongest adaptive attack scenario, they vary the portion k_D used by the defense and evaluate the cases where the adaptive attacker uses a the same/different portion k_A.

However, recent work [39] has re-evaluated temporal dependency frameworks and demonstrated them to be ineffective in detecting adversarial perturbations in the audio domain. The authors of [39] designed attacks that were able to fool the proposed detector in [24] with 100% accuracy, and further report that the adaptive evaluations conducted in [24] are incomplete. In the adaptive attack designed by [39], the CTC loss function used by the attacker incorporates different values of k_A and is therefore able to bypass the temporal dependency detector with minimal added perturbation to audio.

Aside from proposing the temporal-dependency defense for detection, the authors of [24] also study the effectiveness of various input transformation functions in recovering the original transcription from the adversarial counterpart. To this end, they perform experiments with transformation functions such as quantization, down-sampling, local smoothing and auto-encoder reformation of signals. They report that these methods are ineffective in recovering the correct transcription of audio signals. In our work, we will evaluate some of these transformations for the goal of detecting adversarial examples as opposed to recovering benign examples. However, we report that for some attack types, most transformation based defenses are able to recover the benign audio transcription with low CER.

Rajaratnam *et al.* [23] also studied the use of pre-processing techniques such as audio compression, band-pass filtering, audio panning and speech coding as a part of both isolated and ensemble methods for detecting adversarial audio examples generated by a single targeted attack [38]. While they report high detection performance against the targeted adversarial attack proposed by [38], their techniques were not evaluated in an adaptive attack setting and therefore do not provide security guarantees against a future adversary. Given the difficulty of performing defense evaluations, in our work, we perform additional experiments with various input transformation functions to validate or refute the security claims made in existing papers.

3 Methodology

3.1 Threat Model

Adversarial attacks in the audio domain can be classified broadly into two categories: *targeted* and *untargeted* attacks. In targeted attacks the goal of the adversary is to add a small perturbation to an audio signal such that it causes the victim ASR to transcribe the audio to a given target phrase. In untargeted attacks the goal is simply to cause significant error in transcription of the audio signal so that the original transcription cannot be deciphered.

The common goal across both targeted and untargeted attack is to cause mis-transcription of the given speech signal

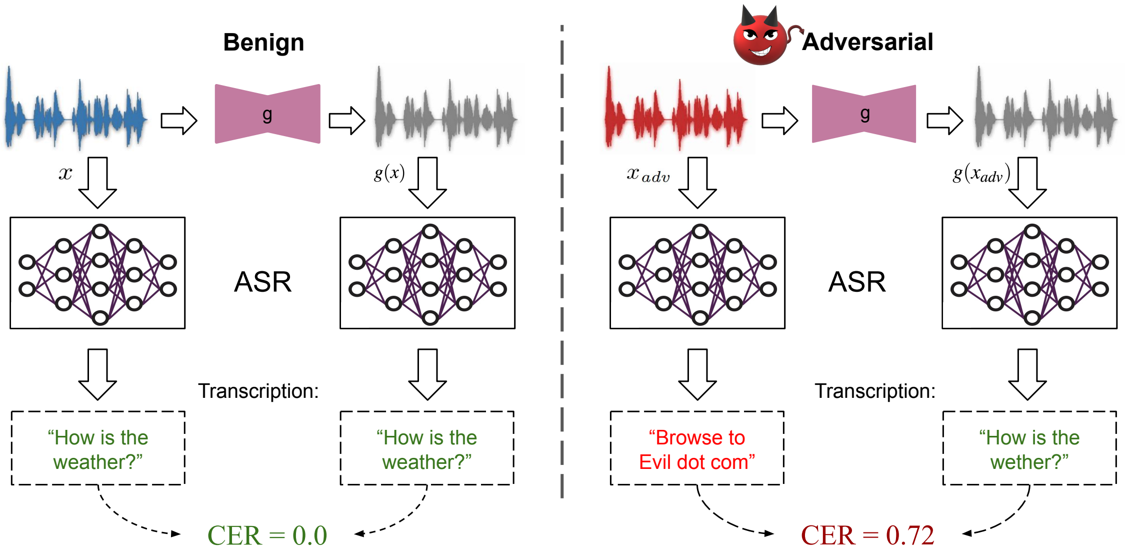


Figure 3: WaveGuard Defense Framework: We first processes the input audio x using an audio transformation function g to obtain $g(x)$. Then the ASR transcriptions or x and $g(x)$ are compared. An input is classified as *adversarial* if the difference between the transcriptions of x and $g(x)$ exceeds a particular threshold.

while keeping the perturbation imperceptible. Therefore, we define an audio adversarial example x_{adv} as a perturbation of an original speech signal x such that the Character Error Rate (CER) between the transcriptions of the original and adversarial examples from an ASR C is greater than some threshold t. That is,

$$CER(C(x), C(x_{adv})) > t \tag{1}$$

and the distortion between x_{adv} and x is constrained under a distortion metric δ as follows:

$$\delta(x, x_{adv}) < \varepsilon. \tag{2}$$

Here, $CER(x,y)$ is the edit distance [43] between the strings x and y normalized by the length of the strings i.e.,

$$CER(x,y) = \frac{EditDistance(x,y)}{max(length(x), length(y))}. \tag{3}$$

L_p norms are popularly used to quantify the distortion δ between the original and adversarial example in the image domain. Following prior works [11, 15] on audio adversarial attacks, we use an L_∞ norm on the waveforms to quantify the distortion between the adversarial and the original signal.

3.2 Defense Framework

The goal of our defense is to correctly detect adversarially modified inputs. The underlying hypothesis for our defense framework is that the network predictions for adversarial examples are often unstable and small changes in adversarial inputs can cause significant changes in network predictions. In the image domain, it has been shown that several input transformation techniques [18–21] such as JPEG compression, randomized smoothing and feature squeezing can render adversarial perturbations ineffective. This is because such input transformations introduce an additional perturbation in the input that can dominate the carefully added adversarial perturbation. On the other hand, predictions for the original (benign) inputs are usually robust to small random perturbations in the input.

Based on this hypothesis, we propose the following defense framework for detecting audio adversarial examples: For a given audio transformation function g, input audio x is classified as adversarial if there is significant difference between the transcriptions $C(x)$ and $C(g(x))$:

$$d(C(x), C(g(x))) > t \tag{4}$$

where d is some distance metric between the two given texts and t is a detection threshold. In our work we use the Character Error Rate (CER) as the distance metric d. z An overview of the defense is depicted in Figure 3. Note that unlike [24], the goal using an input transformation g is not to recover the original transcription of an adversarial example, but to detect if an example is adversarial or benign by observing the difference in the transcriptions of x and $g(x)$.

In this work, we study various input transformation functions g as candidates for our defense framework. We evaluate our defense against four recent adversarial attacks [14, 15, 38] on ASR systems. One of the main insights we draw from our experiments is that in the non-adaptive attack setting, most audio transformations can be effectively used in our defense framework to accurately distinguish adversarial and benign inputs. This result is consistent with the success of input-transformation based defenses in the image domain.

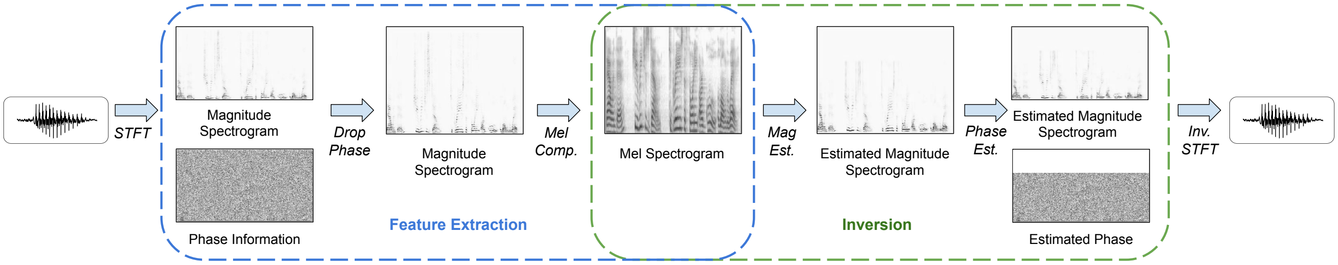


Figure 4: Steps involved in the Mel extraction and inversion transform (Section 4.4). In the extraction step, the phase information of the signal is discarded and the magnitude spectrogram is compressed to a Mel spectrogram using a linear transform. In the inversion step, the waveform is estimated by first estimating the magnitude spectrogram, followed by phase estimation and finally an inverse STFT.

However, in order to use a defense reliably in practice, the defense must be secure against an adaptive adversary who has knowledge of the defense. For an adaptive attack setting, we find that certain input transformations are more robust to attacks than others. Particularly, the transformations which compress audio to perceptually informed representations cannot be easily bypassed even when the attacker has complete knowledge of the defense. This finding is in contrast to the image domain where most input transformation based defenses have been shown to be broken under robust or adaptive adversarial attacks. We elaborate on our adaptive attack scenario and the results in Section 7 and Section 8.

4 Input-transformation functions

We study the following audio transformations as candidates for the input transformation function g:

4.1 Quantization-Dequantization

Several works in the image domain [21, 44, 45], have used quantization based defenses to neutralize the effect of adversarial perturbations. Since adversarial pertubations to audio have small amplitudes, quantization can help reomve added perturbations. In this study, we employ quantization-dequantization in our defense framework, where each waveform sample is quantized to q bits and then reconstructed back to floating point to produce the output approximation of the original input data.

4.2 Down-sampling and Up-sampling

Discarding samples from a waveform during down-sampling could remove a significant portion of the adversarial perturbation, thereby disrupting an attack. To study this effect, we down-sample the original waveform (16 kHz in our experiments), to a lower sampling rate and then estimate the waveform at its original sampling rate using interpolation. We perform this study for a number of different down-sampling rates to find an optimal range of sampling rates for which the defense is effective.

4.3 Filtering

Filtering is commonly applied for noise cancellation applications such as removing background noise from a speech signal. It is intuitive to study the effect of filtering in order to remove adversarial noise from a speech signal. In this work, we use low-shelf and high-shelf filters to clean a given signal. Low-shelf and high-shelf filters are softer versions of high-pass and low-pass filters respectively. That is, instead of completely removing frequencies above or below some thresholds, shelf filters boost or reduce their amplitude. For noise removal, we use a low-shelf filter to reduce the amplitude of frequencies below a threshold and a high-shelf filter to reduce the amplitude of frequencies above a threshold.

In our experiments we first compute the spectral centroid of the audio waveform: Each frame of a magnitude spectrogram is normalized and treated as a distribution over frequency bins, from which the mean (centroid) is extracted per frame. We then compute the median centroid frequency (C) over all frames and set the high-shelf frequency threshold as $1.5 \times C$ and low-shelf frequency threshold as $0.1 \times C$. We then reduce the amplitude of frequencies above and below the respective thresholds using a negative gain parameter of -30.

4.4 Mel Spectrogram Extraction and Inversion

Mel spectrograms are popularly used as an intermediate audio representation in both text-to-speech [46–48] and speech-to-text [49, 50] systems. While reduction of the waveform to a Mel spectrogram is a lossy compression, the Mel spectrogram is a perceptually informed representation that mostly preserves the audio content necessary for speech recognition systems. We use the following Mel spectrogram extraction and inversion pipeline for disrupting adversarial perturbations in our experiments:

Extraction: We first decompose waveforms into time and frequency components using a Short-Time Fourier Transform (STFT). Then, the phase information is discarded from the complex STFT coefficients leaving only the magnitude spectrogram. The linearly-spaced frequency bins of the resultant spectrogram are then compressed to fewer bins which are equally-spaced on a logarithmic scale (usually the Mel scale [51]). Finally, amplitudes of the resultant spectrogram are made logarithmic to conform to human loudness perception, then optionally clipped and normalized to obtain the Mel spectrogram.

Inversion: To invert the Mel spectrogram into a listenable waveform, the inverse of each extraction step is applied in reverse. First, logarithmic amplitudes are converted to linear ones. Then the magnitude spectrogram is estimated from the Mel spectrogram using the approximate inverse of the Mel transformation matrix. Next, the phase information is estimated from the magnitude spectrogram using a heurisitc algorithm such as Local Weighted Sum (LWS) [52] or Griffin Lim [53]. Finally, the inverse STFT is used to render audio from the estimated magnitude spectrogram and phase information.

We hypothesize that reconstructing audio from a perceptually informed representation can potentially remove the adversarial perturbation while preserving the speech content that is perceived by the human ear. While some speech recognition systems also use Mel spectrogram features, we find that reconstructing audio from the *compressed* Mel spectrograms introduces enough distortion in the original waveform, such that the ASR Mel features of the newly reconstructed audio are different from the original audio. The distortion in the reconstructed audio is introduced by the magnitude estimation and phase estimation steps depicted in Figure 4. In order to bypass a defense involving Mel extraction and inversion, an adaptive attacker will need to craft a perturbation that can be retained in the compressed Mel spectrogram representation, making it challenging to keep the perturbation imperceptible. In our adaptive attack experiments in Section 8 we demonstrate that even when the attacker uses a differentiable implementation of the Mel extraction and inversion pipeline, it cannot easily be bypassed without introducing a clearly perceptible adversarial noise in the signal.

4.5 Linear Predictive Coding

Linear Predictive Coding (LPC) is a speech encoding technique that uses a source-filter model based on a mathematical approximation of the human vocal tract. The model assumes that a source signal $e(n)$ (which models the vocal chords) is passed as input to a resonant filter $h(n)$ (that models the vocal tract) to produce the resultant signal $x(n)$. That is:

$$x(n) = h(n) * e(n) \tag{5}$$

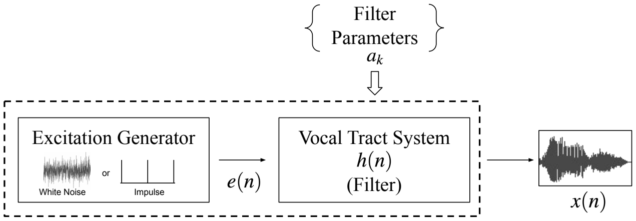


Figure 5: Model for linear predictive analysis of speech signals.

The source excitation $e(n)$ can either be quasi-periodic impulses (during voiced speech) or random noise (during unvoiced speech). Both these source excitation sources are spectrally flat implying that all spectral information is modeled in the filter parameters.

LPC assumes a p^{th} order all-pole filter $h(n)$ which means that each waveform sample is modelled as a linear combination of p previous values. That is,

$$x(n) = \Sigma_{k=1}^{k=p} a_k x(n-k) + e(n). \tag{6}$$

The basic problem of LPC analysis is to estimate the filter parameters a_k. Since the source signal is assumed to be an impulse train or random white noise, the problem is formulated as minimizing $||e(n)||^2$ which is the power of the excitation signal. This reduces the parameter-estimation problem to a linear regression problem in which the goal is to minimize:

$$\textit{minimize:}\ \langle ||e(n)||^2 \rangle = \langle (x(n) - \Sigma_{k=1}^{k=p} a_k x(n-k))^2 \rangle \tag{7}$$

Here, $\langle\rangle$ denotes averaging over finite number of waveform samples. In practice, a long time-varying signal is divided into overlapping windows of size w and LPC coeffecients a_k are estimated for each window by solving the above linear regression problem. To re-synthesize the signal from the estimated coefficients, we use a random-noise excitation signal. In our experiments, we use 25 millisecond windows with 12.5 millisecond overlap. We experiment with different numbers of the LPC coeffecients which control the compression level of the original signal.

Since LPC models the human vocal tract system, it preserves the phonetic information of speech in the filter parameters. Bypassing a defense involving LPC transform, would require the adversary to add an adversarial perturbation that can be preserved in the LPC filter coeffecients; thereby requiring the adversary to modify the phonetic information in speech. We empirically demonstrate that the LPC transform cannot be easily bypassed by an adaptive adversary.

5 Experimental Setup

We evaluate our defense against the following recent audio adversarial attacks on speech recognition systems [11, 14, 15]:

- **Carlini:** Attack introduced in [11]. This is a white-box targeted attack on the Mozilla Deepspeech [4] ASR system, where the attacker trains an adversarial perturbation by minimizing the CTC loss between the target transcription and the ASR's prediction. This attack minimizes the L_∞ norm of the adversarial perturbation to constrain the amount of distortion.
- **Qin-I:** Imperceptible attack described in [14]. This is another white-box targeted attack that focuses on ensuring imperceptibility of the adversarial perturbation by using psycho-acoustic hiding. The victim ASR for this attack is Google Lingvo [3].
- **Qin-R:** Robust attack described in [14]. This attack incorporates input transformations during training of the adversarial perturbation which simulate room environments. This improves the attack robustness in real world settings when played over the air. The victim ASR for this attack is Google Lingvo [3].
- **Universal:** We implement the white-box attack described in [15]. This is an untargeted attack which finds an input-agnostic perturbation that can cause significant disruption in the transcription of the adversarial signal. In our work, we follow the algorithm provided by the authors and craft universal perturbation with an L_∞ bound of 400 (for 16-bit audio wave-forms with sample values in the range -32768 to 32768). The victim ASR for this attack is Mozilla DeepSpeech [4].

Target Adversarial Commands
"browse to evil dot com"
"hey google cancel my medical appointment"
"hey google"
"this is an adversarial example"

Table 1: Adversarial commands used for constructing targeted adversarial examples.

5.1 Dataset and Attack Evaluations

We conduct all our experiments on the Mozilla Common Voice dataset, which contains 582 hours of audio across 400,000 recordings in English. The audio data is sampled at 16 kHz. We evaluate on the same subset of the Mozilla Common Voice dataset, as used in [11], that is, the first 100 examples from the Mozilla Common Voice test set. We construct adversarial examples on this dataset using each of the attacks described above. In the targeted attack scenario, we randomly choose one of the target phrases listed in Table 1 and follow the attack algorithms to create 100 pairs of original and adversarial examples for each attack type. For the untargeted universal attack, we train the universal perturbation on the same subset of Mozilla Common Voice examples with L_∞ distortion bound of 400.

Attack evaluations: We achieve 100% attack success rate for *Carlini* and *Qin-I* attacks. For *Qin-R*, the attack achieves 47% success rate (similar to that reported in the paper [14]) on 100 examples. In our experiments when recreating the *Universal* attack, we achieve an attack success rate of 81% using the same criteria as described in [15] i.e., the attack is considered successful when the CER between original and adversarial transcriptions is greater than 0.5.

5.2 Evaluation Metrics

As described in Section 3.2, in our detection framework, we label an example as *adversarial* or *benign* based on the CER between x and $g(x)$. The decision threshold t controls the true positive rate and false positive rate of our detector. Following standard procedure to evaluate such detectors [24], we calculate the *AUC score* - Area Under the ROC curve. A higher AUC score indicates that the detector has more discriminative power against adversarial examples.

Additionally, we also report the *Detection Accuracy* which is calculated by finding the best detection threshold t on a separate set containing 50 adversarial and benign examples.

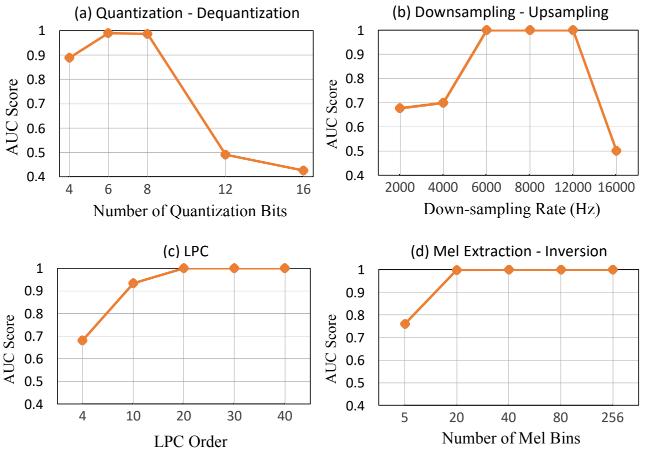


Figure 6: Detection AUC Scores against *Carlini* attack at varying compression levels for the following transforms: (a) Quantization - Dequantization; (b) Downsampling - Upsampling; (c) Linear Predictive Coding (LPC); and (d) Mel Spectrogram Extraction- Inversion.

6 Evaluation against Non Adaptive Attacks

The various input transformation functions we consider can be parameterized to control the compression level of the transformation. There is a trade-off between the compression level and the discriminative power of the detector. At low compression levels the transformation may not eliminate the adversar-

		AUC Score				*Detection Accuracy*			
Defense	Hyper-params	Carlini	Universal	Qin-I	Qin-R	Carlini	Universal	Qin-I	Qin-R
Downsampling - Upsampling	6000 kHz	1.00	0.91	1.00	1.00	100%	88%	100%	100%
Quantization - Dequantization	6 bits	0.99	0.92	1.00	0.93	98.5%	88%	99%	95%
Filtering	(Section 4.3)	1.00	0.92	1.00	1.00	99.5%	86%	100%	100%
Mel Extraction - Inversion	80 Mel-bins	1.00	0.97	1.00	1.00	100%	92%	100%	100%
LPC	LPC order 20	1.00	0.91	1.00	1.00	100%	83%	100%	100%

Table 2: Evaluations for each input transformation defense against various non-adaptive attacks. We use two objective metrics: AUC score and Attack Detection Accuracy for evaluation (higher values are better for both metrics).

ial perturbation. In contrast, at very high compression levels, even the benign signals may become significantly distorted causing substantial change in their transcriptions. Keeping this in mind, we perform a search over the hyper-parameters for different audio transforms. The AUC score of the detector against the *Carlini* attack for different transformation functions at varying compression levels is depicted in Figure 6. For most transformations, we observe the expected pattern where the defense is effective at some optimal compression levels and the AUC falls at very high or low compression levels. The Mel extraction-inversion pipeline is effective for a wide range of *Mel-bins* possibly due to the distortion introduced by the phase estimation step during the inversion stage. For the *Filtering* transform we do not perform a hyper-parameter search and use the transformation parameters described in Section 4.3.

6.1 Detection Scores

Based on the above described search, we find the optimal hyper-parameters for each of the transforms and report the detection scores against all the attacks in Table 2. We observe that at optimal compression levels, all the input transforms listed in Section 4 can achieve high discriminative performance against adversarial examples. As compared to targeted adversarial examples, it is harder to detect examples with universal adversarial perturbations. This is because universal perturbations attempt to distort the original transcription rather than targeting a very different phrase. Interestingly, we find that the defense is effective even against the *Qin-R* attack which incorporates noise simulation during training and leads to adversarial examples that are robust to small changes. We elaborate on this result in the following Section.

6.2 Analysis of undefended and defended transcriptions

In Figure 7 we provide comparisons of Mean CER between transcriptions of audio before and after passing through a given transformation function (g) for both benign ($orig$) and adversarial examples (adv). Additionally, we also calculate the CER between the transcriptions of the defended adversarial example and its benign counterpart: $CER(orig, g(adv))$.

The discriminative power of the detector is indicated by the difference between $CER(orig, g(orig))$ (blue) and $CER(adv, g(adv))$ (red). A high difference between the red and blue bar graphs in Figure 7 indicates easier detection of adversarial examples. From these results we can observe that detecting the *Qin-I* attack is easier than detecting the *Carlini* [11] attack. We can further deduce that detecting *Universal* attacks is generally more difficult for any given transformation function compared to the *Carlini* and *Qin-I* attacks.

The metric $CER(orig, g(adv))$ helps evaluate the ability of the transformation function to recover the original transcript from the adversarial audio. A low $CER(orig, g(adv))$ indicates better recovery of the original transcript. We find that for the imperceptible attack *Qin-I*, the recovery rate of the original transcript is higher than any other attack indicating that the adversarial perturbation is unstable to small changes in inputs.

The *Qin-R* attack has a lower $CER(adv, g(adv))$ for most transformations as compared to *Qin-I* which suggests that the adversarial perturbation generated by the *Qin-R* attack is relatively more robust to input transformations. Also, recovering the original transcription is much harder as compared to *Qin-I* and is indicated by higher $CER(orig, g(adv))$ values. However, there is still a significant difference between the blue and red bar graphs for *Qin-R*, which can be used to discriminate between adversarial and benign samples. This result is consistent with the high detection accuracy reported in Table 2, since the transformations are successful in disrupting the adversarial perturbations.

We provide a few sample transcriptions from our experiments in Figure 8. The green commands indicate the transcriptions from benign audio samples, while the red transcriptions refer to adversarial commands from each attack type. Overall, the results in Figure 7 and Figure 8 demonstrate that the ability to recover benign commands is dependent on the type of attack and varies for each input transformation function.

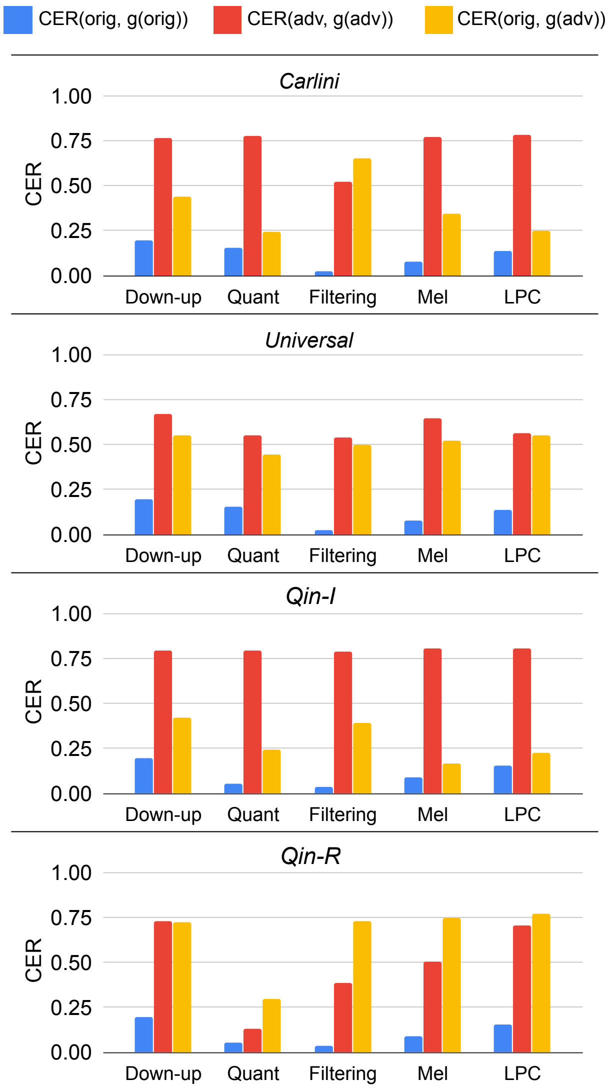


Figure 7: Mean Character Error Rate (CER) between the ASR transcriptions of un-transformed (x) and transformed ($g(x)$) audio. *CER(orig,g(orig))* and *CER(adv, g(adv))* indicate the CER between transcriptions of x and $g(x)$ for benign and adversarial samples respectively. *CER(orig, g(adv))* is the CER between the defended adversarial signal and its benign counterpart.

6.3 Timing analysis

To implement our defense framework in practice, we have to perform two forward passes through our ASR model to obtain the transcriptions $C(x)$ and $C(g(x))$. It is ideal to parallelize these two forward passes, so that the only computational overhead introduced by the defense is that of the transformation function g. Table 3 provides the average Wall-Clock time in seconds of each transformation function averaged over the 100 audio files (entire test set). Since some of our transformation functions were implemented solely on CPU, we provide timing comparisons for all implementations on the Intel Xeon CPU platform. The average inference time over the test set for Mozilla Deepspeech ASR model is 2.540 seconds and that of Google Lingvo ASR model is 4.212 seconds on the Intel Xeon CPU Platform.

Process	Avg. Wall-Clock time (s)
Deepspeech ASR	2.540
Lingvo ASR	4.212
Downsampling-Upsampling	0.148
Quantization-Dequantization	0.001
Filtering	0.035
Mel Extraction - Inversion	0.569
LPC	0.781

Table 3: Average Wall-Clock time in seconds required for transcription of audio by ASR models and each transformation function on Intel Xeon CPU platform. The Wall-Clock time is averaged over the entire test set.

7 Adaptive Attack

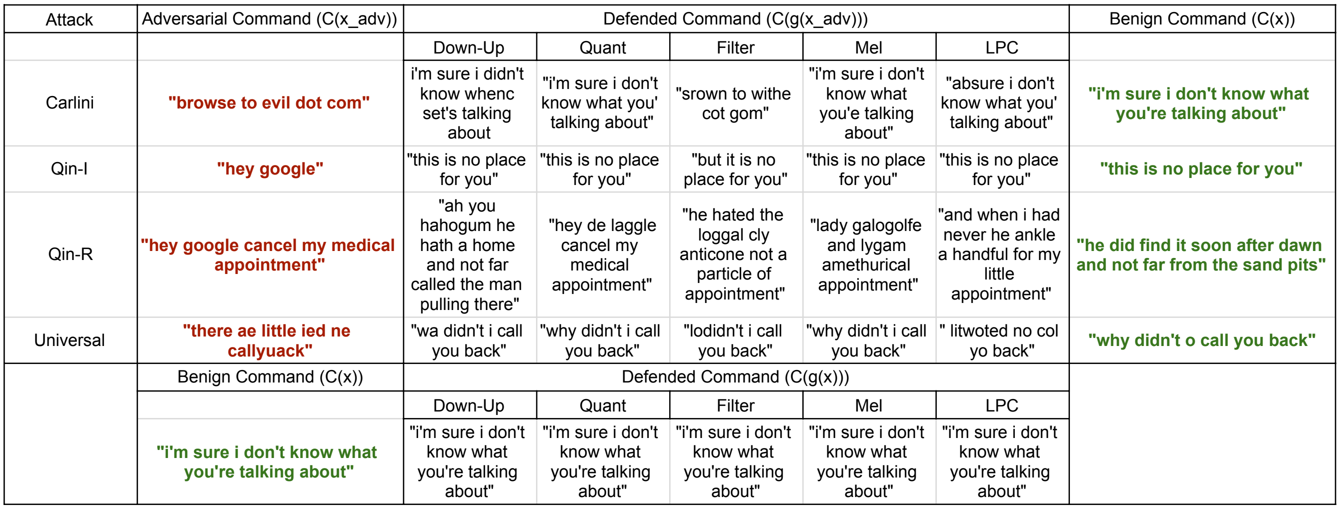
While our defense framework can accurately discriminate adversarial from benign examples for existing attacks, it only offers security in a "zero-knowledge" attack scenario where the attacker is not aware of the defense being present. As motivated in Section 2.2, in order to use our defense framework reliably in practice, it is important to evaluate it against an adaptive adversary who has complete knowledge of the defense and intend to design a perturbation that can bypass the defense mechanism.

In the adaptive attack setting, we will focus on the more impactful targeted attack scenario, where the adversary designs an adversarial perturbation that causes the victim ASR system to transcribe the input audio into a specific target phrase. In order to bypass the proposed defense framework, the adversary must craft an adversarial perturbation such that the transcription of $C(x_{adv})$ and $C(g(x_{adv}))$ match closely with each other and the target transcription τ. Therefore, to craft such a perturbation δ, the adversary aims to optimize the following problem:

$$\textit{minimize:}\ \ |\delta|_\infty + c_1 \cdot \ell(x+\delta,\tau) + c_2 \cdot \ell(g(x+\delta),\tau)$$

where, $\ell(x',t) = \text{CTC-Loss}(C(x'),t)$ and c_1 and c_2 are hyperparameters that control the weights of the respective loss terms. Since optimization process over the L_∞ metric is often unstable [11], we modify our optimization objective as follows:

$$\begin{aligned} \textit{minimize:}\ \ & c \cdot |\delta|_2^2 + c_1 \cdot \ell(x+\delta,\tau) + c_2 \cdot \ell(g(x+\delta),\tau) \\ & \textit{such that}\ |\delta|_\infty < \varepsilon \end{aligned} \tag{8}$$



Attack	Adversarial Command (C(x_adv))	Defended Command (C(g(x_adv)))					Benign Command (C(x))
		Down-Up	Quant	Filter	Mel	LPC	
Carlini	"browse to evil dot com"	i'm sure i didn't know whenc set's talking about	"i'm sure i don't know what you' talking about"	"srown to withe cot gom"	"i'm sure i don't know what you'e talking about"	"absure i don't know what you' talking about"	"i'm sure i don't know what you're talking about"
Qin-I	"hey google"	"this is no place for you"	"this is no place for you"	"but it is no place for you"	"this is no place for you"	"this is no place for you"	"this is no place for you"
Qin-R	"hey google cancel my medical appointment"	"ah you hahogum he hath a home and not far called the man pulling there"	"hey de laggle cancel my medical appointment"	"he hated the loggal cly anticone not a particle of appointment"	"lady galogolfe and lygam amethurical appointment"	"and when i had never he ankle a handful for my little appointment"	"he did find it soon after dawn and not far from the sand pits"
Universal	"there ae little ied ne callyuack"	"wa didn't i call you back"	"why didn't i call you back"	"lodidn't i call you back"	"why didn't i call you back"	" litwoted no col yo back"	"why didn't o call you back"
	Benign Command (C(x))	Defended Command (C(g(x)))					
		Down-Up	Quant	Filter	Mel	LPC	
	"i'm sure i don't know what you're talking about"	"i'm sure i don't know what you're talking about"	"i'm sure i don't know what you're talking about"	"i'm sure i don't know what you're talking about"	"i'm sure i don't know what you're talking about"	"i'm sure i don't know what you're talking about"	

Figure 8: Sample transcriptions of un-transformed(x) and transformed audio($g(x)$) for both benign and adversarial examples.

7.1 Gradient Estimation for Adaptive Attack

To solve the optimization problem given by equation 8 using gradient descent, the attacker must back-propagate the CTC-Loss through the ASR model and the input transformation function g. In case a differentiable implementation of g is not available, we use the Backward Pass Differentiable Approximation (BPDA) technique [6] to craft adversarial examples. That is, during the forward pass we use the exact implementation of the transformation function as used in our defense framework. During the backward pass, we use an approximate gradient implementation of the transformation g. We first perform the adaptive attack using the straight-through gradient estimator [6]. That is, we assume that the gradient of the loss with respect to the input x to be the same as the gradient of the loss with respect to $g(x)$:

$$\nabla_x \ell(g(x))|_{x=\hat{x}} \approx \nabla_x \ell(x)|_{x=g(\hat{x})}. \tag{9}$$

In our experiments, we find that the straight-through estimator is effective in breaking the Quantization-Dequantization and Filtering transformation functions at low perturbation levels. However, using a more accurate gradient estimate can lead to a stronger attack. Specifically for the Mel extraction-inversion and LPC transformations, we find that using a straight-through gradient estimator does not work for solving the above optimization problem (Equation 8). We discuss our results of using a straight-through gradient estimator for LPC transform in Appendix D.. Also, using a straight-through estimator for the Downsampling-Upsampling transform results in high distortion for adversarial perturbations. Therefore, we implement differentiable computational graphs for the following three transforms in TensorFlow:

Downsampling-Upsampling: We use TensorFlow's bilinear resizing methods to first downsample the audio to the required sampling rate and then re-estimate the signal using bi-linear interpolation.

Mel Extraction - Inversion: For the Mel extraction-inversion transform we use TensorFlow's STFT implementation to obtain the magnitude spectrogram, then perform the Mel transform using matrix multiplication with the Mel basis, and estimate the waveform using the iterative Griffin-Lim [53] algorithm implemented in TensorFlow [54].

LPC transform: We implement the LPC analysis and synthesis process in TensorFlow. Specifically, for each window in the original waveform, we first estimate LPC coefficients by solving the linear regression problem given by Equation 7. Next, for the reconstruction process, we generate the residual excitation signal using the exact same implementation as used in our defense. We also fix the random seed of the excitation generator in both our defense and our adaptive attacks for a complete knowledge white box attack scenario. Next, we implement auto-regressive filtering of the residual signal with the LPC coefficients for that window to synthesize the signal for the given window. Finally, we add and combine the filtered signal for each overlapping window to generate the transformed audio.

Note that for all the adaptive attacks, we use the original defense implementations in the forward pass and use the differentiable implementation only during the backward pass.

7.2 Adaptive Attack Algorithm

Algorithm 1 details our adaptive attack implementation. We closely follow the targeted attack implementation in [11] and incorporate the optimization objective of our adaptive attack specified by Equation 8 and BPDA. We choose $c_1 = c_2 = 1$ since both loss terms have the same order of magnitude. Following the default open source implementation of [11], we do not penalize L_2 distortion. We optimize for 5000 iterations and use a learning rate of 10. Any time the attack succeeds, we re-scale the perturbation bound by a factor of 0.8 to encourage less distorted (quieter) adversarial examples. We include the exact implementation of the adaptive attack and the differentiable computational graphs for BPDA in our code.[2]

Algorithm 1 Adaptive attack algorithm

```
Initialize $rescaleFactor \leftarrow 1$
Initialize $\delta \leftarrow 0$
Initialize $bestDelta \leftarrow null$
for $iterNum$ in 1 to $MaxIters$ do
    $loss \leftarrow c \cdot |\delta|_2^2 + c_1 \cdot \ell(x+\delta, t) + c_2 \cdot \ell(g(x+\delta), t)$
    $\nabla\delta \leftarrow BPDA(loss, \delta)$
    $\delta \leftarrow \delta - \alpha\, sign(\nabla\delta)$
    $\delta \leftarrow rescaleFactor * clip_\varepsilon(\delta)$
    if $C(x+\delta) = C(g(x+\delta)) = \tau$ then
        $bestDelta \leftarrow \delta$
        $rescaleFactor \leftarrow rescaleFactor \times 0.8$
if $bestDelta$ is $null$ then
    $bestDelta \leftarrow \delta$
return $(x + bestDelta)$
```

8 Adaptive Attack Evaluation

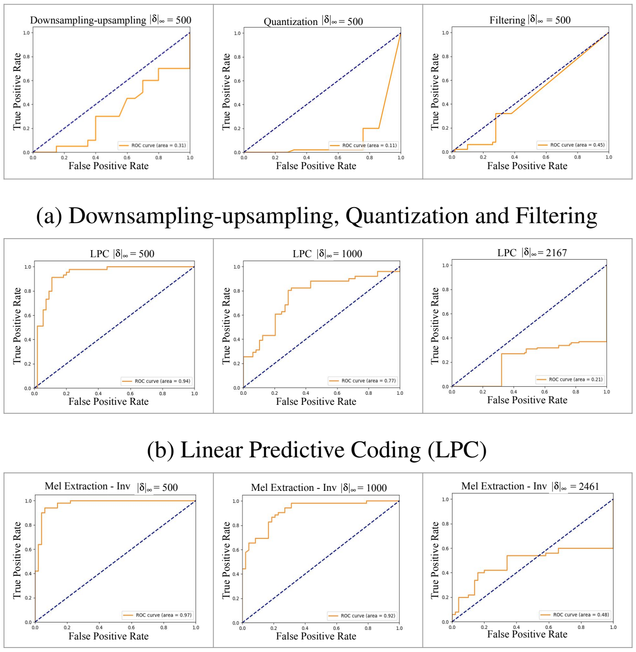
In this section, we test the limits of our defense and evaluate the breaking point for each transformation function through adaptive attacks in white box setting. We conduct adaptive attack evaluations on the same dataset used in our previous experiments. The victim ASR for the adaptive attack is the Mozilla DeepSpeech model. In order to evaluate the imperceptibility of adversarial perturbations, we quantify the distortion of adversarial perturbations as follows.

Distortion Metrics and Relative Loudness: We first implement adaptive attacks using an initial distortion bound $|\varepsilon|_\infty = 500$. Note that we are using a 16-bit waveform representation which means that the waveform samples are in the range -32768 to 32768. An L_∞ distortion of 500 is fairly perceptible although it does not completely mask the original signal.[3] Along with the L_∞ norm of the perturbation, we report another related metric $dB_x(\delta)$ [11, 15] that measures the relative loudness of the perturbation with respect to the original signal in Decibels(dB). The metric $dB_x(\delta)$ is defined as follows:

$$\begin{aligned} dB(x) &= max_i 20\log_{10}(x_i) \\ dB_x(\delta) &= dB(\delta) - dB(x) \end{aligned} \tag{10}$$

The more negative $dB_x(\delta)$ is, the quieter is the adversarial perturbation. For comparison, -31 dB is roughly the difference between ambient noise in a quiet room and a person talking [11]. While we start with an initial L_∞ (ε_∞) bound of 500 in our experiments, the final distortion norm (δ_∞) can be much smaller than the initial bound. This is because our optimization objective penalizes high distortion amounts and our algorithm re-scales the perturbation bound by a factor of 0.8 every time the attack succeeds.

Generally, prior work on attacks to ASR systems apply particular attention to minimize perturbation distortions, in order to encourage imperceptibility of adversarial audio. Towards this goal of generating imperceptible adversarial examples, Qin et al. [14] and Universal [15] generate examples with maximum allowed distortion of L_∞ = 400, while Carlini et al. [11] generate examples with maximum distortion of L_∞ = 100. However for conducting our adaptive attack evaluation, since we aim to test the breaking point of each transformation function, we generate adversarial perturbations at much higher L_∞ bounds (500, 1000, 4000) that are significantly more audible to the human ear.



(a) Downsampling-upsampling, Quantization and Filtering

(b) Linear Predictive Coding (LPC)

(c) Mel Extraction - Inversion

Figure 9: Detection ROC curves for different transformation functions against adaptive attacks (Section 8) with various magnitudes of adversarial perturbation ($|\delta|_\infty$).

Table 4 presents the results for our adaptive attack against various input transformation functions. We provide the Receiver Operating Characteristic (ROC) of the detector in the adaptive attack settings for different transformation functions under different magnitudes of perturbation in Figure 9. A *true positive* implies an example that is adversarial and is correctly identified as adversarial. We evaluate the adaptive attacks on two aspects: 1) *Attack Performance:* How successful was the

[2] Code: https://github.com/waveguard/waveguard_defense
[3] Audio Examples: https://waveguard.herokuapp.com

	Distortion metrics			*Attack Performance*				*Detection Scores*	
Defense	ϵ_∞	$\lvert\delta\rvert_\infty$	$dB_x(\delta)$	SR (x_{adv})	SR ($g(x_{adv})$)	$CER(x_{adv},\tau)$	$CER(g(x_{adv}),\tau)$	AUC	Acc.
None	500	81	-45.3	100%	-	0.00	-	-	-
Downsampling - Upsampling	500	**342**	-32.7	100%	78%	0.00	0.05	0.31	50.0%
Quantization - Dequantization	500	**215**	-36.7	100%	81%	0.00	0.01	0.11	50.0%
Filtering	500	**92**	-44.1	91%	72%	0.01	0.02	0.45	50.0%
Mel Extraction - Inversion	500	500	-29.4	34%	0%	0.11	0.44	0.97	95.5%
LPC	500	500	-29.4	43%	0%	0.06	0.51	0.94	86.0%
Mel Extraction - Inversion	1000	1000	-23.5	53%	0%	0.05	0.34	0.92	84.0%
LPC	1000	1000	-23.5	72%	0%	0.01	0.29	0.77	72.5%
Mel Extraction - Inversion	4000	**2461**	-15.1	100%	31%	0.00	0.08	0.48	50.0%
LPC	4000	**2167**	-16.7	100%	73%	0.0	0.03	0.21	50.0%

Table 4: Adaptive attack evaluations against different transformation functions. ϵ_∞ is the initial L_∞ bound used in the attack algorithm and δ_∞ is the mean L_∞ norm of the perturbations obtained after applying the adaptive attack algorithm. Bolded values indicate the δ_∞ required to completely break (AUC $\leq$ 0.5) a particular transformation function based defense. $dB_x(\delta)$ is the relative loudness of the perturbation with respect to the examples in the dataset (the lower the quieter). SR (x_{adv}) and SR $(g(x_{adv}))$ indicate the attack success rate for un-transformed (x_{adv}) and transformed audio $(g(x_{adv}))$ respectively obtained using the adaptive attack algorithm on a given transformation function.

adaptive attack in its objective? 2) *Detection Scores:* How effective is our detector for the adversarial audios generated by the attack?

For the adaptive attacks against the *Downsampling-upsampling*, *Quantization-Dequantization* and *Filtering* transforms, we achieve low CER between the target transcription and transcriptions for x_{adv} and $g(x_{adv})$ ($CER(x_{adv},\tau)$ and $CER(g(x_{adv}))$ respectively). This makes it harder for the detector to discriminate between adversarial and benign samples thereby resulting in a drastic drop in detector AUC and accuracy scores as compared to the non-adaptive scenario. Amongst these three transformations, bypassing *Downsampling-upsampling* requires the highest amount of perturbation ($\delta_\infty = 342$) indicating that it serves as a more robust defense transformation as compared to *Quantization-Dequantization* and *Filtering*. The columns $SR(x_{adv})$ and $SR(g(x_{adv}))$ indicate the percentage of examples that transcribed exactly to the target phrase for the un-transformed and transformed adversarial inputs respectively.

The calibration of the detection threshold depends on the use case of the ASR system—for a user facing ASR system, the number of legitimate commands would usually be very high as compared to the number of adversarial commands. Therefore, the false positive rate needs to be extremely low for such ASR systems. As shown in Figure 11 (Appendix A.), in the non-adaptive attack scenario, we are able to achieve a very high true positive rate at 0% false positive rate for the targeted adversarial attacks (Carlini and Qin-I) for all transformation functions. Therefore a low detection threshold can be reliable against non-adaptive adversaries and also not interfere with the user experience. In the adaptive attack scenario, while both LPC and Mel inversion achieve higher AUC scores as compared to other transforms, Mel inversion transform gives the highest true positive rate at extremely low false positive rates. Therefore, amongst the transformation functions studied in our work, Mel Extraction and Inversion serves as the best defense choice for user facing ASR systems.

Robustness of perceptually informed representations: For both Mel extraction-inversion and LPC transformations, although we observe a drop in the detector scores as compared to the non-adaptive attack setting, we are not able to completely bypass the defense using the initial distortion bound $\epsilon_\infty = 500$. Note that a perturbation higher than this magnitude, has $dB_x(\delta) > -29$ which is more audible than ambient noise in a quiet room ($dB_x(\delta) = -31$) [38, 55]. In order to test the limit at which the defense breaks, we successively increase the allowed magnitude of perturbation. We are able to completely break the defense (AUC $\leq$ 0.5) at $\delta_\infty = 2479$ and $\delta_\infty = 2167$ for Mel extraction-inversion and LPC transforms respectively. These perturbations are more than 6× higher than that required to break any of the other transformation functions studied in our work and more than 25× higher than that required to fool an undefended model. This suggests that using perceptually informed intermediate representations prove to be more robust against adaptive attacks as compared to naive compression and decompression techniques.

Figure 10 reports the same metrics as those reported in Figure 7 for the adaptive attack scenario with an initial $\epsilon_\infty = 500$. The $CER(adv, g(adv))$ (red bar) drops below $CER(orig, g(orig))$ (blue bar) for *Downsampling-upsampling*, *Quantization-Dequantization* and *Filtering* transforms thereby breaking these defenses. In contrast, the red bar for *Mel extraction-inversion* and *LPC* based defense is much higher than the blue bar indicating that the defense is more robust under this adaptive attack setting.

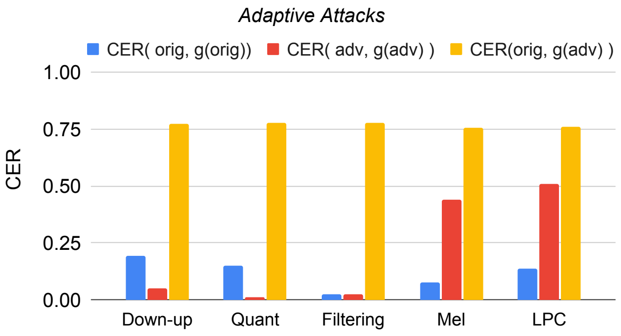


Figure 10: Mean CER between the ASR transcriptions of un-transformed (x) and transformed ($g(x)$) audio for adaptive attacks with an initial distortion $\varepsilon_\infty = 500$.

9 Discussion

Do learnings from adversarial defenses in the image domain transfer over to the audio domain? We find that not all learnings about input-transformation based defenses in the image domain transfer to the speech recognition domain. It has been shown that input-transformation based adversarial defenses can be easily bypassed using robust or adaptive attacks for image classification systems [6, 41]. However, an ASR system is a substantially different architecture as compared to an image classification model. ASR systems operate on time-varying inputs and map each input frame to a language token. Since they rely on Recurrent Neural Networks (RNNs), the token prediction for each frame also depends on other frames in the signal. For targeted attacks, that are robust to a transformation g, we need to find an adversarial example x_{audio} such that both x_{audio} and $g(x_{audio})$ map to the target language tokens across all time-steps. On the other hand, for the image classification problem, the adaptive attack goal is simpler: Find an image x_{image}, such that both x_{image} and $g(x_{image})$ map to the same class label. Therefore, in our adaptive attack experiments, we need to add significant amount of perturbation to bypass the defense even for simple transformation functions. We also find that adversarial attacks targeting undefended ASR models do not transfer to defended models even at high perturbation levels, in contrast to results reported in the image domain [39]. Details of this experiment are provided in Appendix C..

10 Conclusion

We present *WaveGuard*, a framework for detecting audio adversarial inputs, to address the security threat faced by ASR systems. Our framework incorporates audio transformation functions and analyzes the ASR transcriptions of the original and transformed audio to detect adversarial inputs. We demonstrate that WaveGuard can reliably detect adversarial inputs from recently proposed and highly successful targeted and untargeted attacks on ASR systems. Furthermore, we evaluate WaveGuard in the presence of an *adaptive* adversary who has complete knowledge of our defense. We find that only at significantly higher magnitudes of adversarial perturbation, which are audible to the human ear, can an adaptive adversary bypass transformations that compress input to perceptually informed audio representations. In contrast, naive audio transformation functions can be easily bypassed by an adaptive adversary using small inaudible amounts of perturbations. This makes transformations such as LPC and Mel extraction-inversion more robust candidates for defense against audio adversarial attacks.

Acknowledgements

We thank our reviewers for their valuable and comprehensive feedback. This work was supported by SRC under Task ID: 2899.001 and ARO under award number W911NF-19-1-0317.

References

[1] L. R. Rabiner, R. W. Schafer *et al.*, "Introduction to digital speech processing," *Foundations and Trends® in Signal Processing*, 2007.

[2] 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, ICML*, 2016.

[3] J. Shen, P. Nguyen, Y. Wu, Z. Chen, M. X. Chen, Y. Jia, A. Kannan, T. N. Sainath, Y. Cao, and et al., "Lingvo: a modular and scalable framework for sequence-to-sequence modeling," *ArXiv*, vol. abs/1902.08295, 2019.

[4] A. Y. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, and A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *CoRR*, vol. abs/1412.5567, 2014.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *Stat*, 2015.

[6] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, 2018.

[7] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," *2017 IEEE Symposium on Security and Privacy (SP)*, 2017.

[8] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2016.

[9] M. Alzantot, B. Balaji, and M. B. Srivastava, “Did you hear that? adversarial examples against automatic speech recognition,” *CoRR*, vol. abs/1801.00554, 2018. [Online]. Available: http://arxiv.org/abs/1801.00554

[10] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, “Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding,” *arXiv preprint arXiv:1808.05665*, 2018.

[11] N. Carlini and D. Wagner, “Audio adversarial examples: Targeted attacks on speech-to-text,” in *2018 IEEE Security and Privacy Workshops (SPW)*, 2018.

[12] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. Wagner, and W. Zhou, “Hidden voice commands,” in *25th USENIX Security Symposium*, 2016.

[13] H. Yakura and J. Sakuma, “Robust audio adversarial example for a physical attack,” *CoRR*, vol. abs/1810.11793, 2018. [Online]. Available: http://arxiv.org/abs/1810.11793

[14] 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*, 2019.

[15] P. Neekhara, S. Hussain, P. Pandey, S. Dubnov, J. McAuley, and F. Koushanfar, “Universal adversarial perturbations for speech recognition systems,” in *Proc. Interspeech*, 2019.

[16] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, “Commandersong: A systematic approach for practical adversarial voice recognition,” in *27th USENIX Security Symposium*, 2018.

[17] Y. Chen, X. Yuan, J. Zhang, Y. Zhao, S. Zhang, K. Chen, and X. Wang, “Devil’s whisper: A general approach for physical adversarial attacks against commercial blackbox speech recognition devices,” in *29th USENIX Security Symposium*, 2020.

[18] D. Meng and H. Chen, “Magnet: a two-pronged defense against adversarial examples,” in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[19] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, “Countering adversarial images using input transformations,” in *International Conference on Learning Representations, ICLR*, 2018.

[20] J. Lin, C. Gan, and S. Han, “Defensive quantization: When efficiency meets robustness,” *Artificial Intelligence, Communication, Imaging, Navigation, Sensing Systems*, 2019.

[21] F. Khalid, H. Ali, H. Tariq, M. A. Hanif, S. Rehman, R. Ahmed, and M. Shafique, “Qusecnets: Quantization-based defense mechanism for securing deep neural network against adversarial attacks,” in *2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, 2019.

[22] B. Liang, H. Li, M. Su, X. Li, W. Shi, and X. Wang, “Detecting adversarial image examples in deep neural networks with adaptive noise reduction,” *IEEE Transactions on Dependable and Secure Computing*, 2018.

[23] K. Rajaratnam, K. Shah, and J. Kalita, “Isolated and ensemble audio preprocessing methods for detecting adversarial examples against automatic speech recognition,” in *Conference on Computational Linguistics and Speech Processing (ROCLING)*, 2018.

[24] Z. Yang, P. Y. Chen, B. Li, and D. Song, “Characterizing audio adversarial examples using temporal dependency,” in *7th International Conference on Learning Representations, ICLR*, 2019.

[25] D. Iter, J. Huang, and M. Jermann, “Generating adversarial examples for speech recognition,” 2017.

[26] T. Vaidya, Y. Zhang, M. Sherr, and C. Shields, “Cocaine noodles: Exploiting the gap between human and machine speech recognition,” in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, 2015.

[27] L. E. Baum and J. A. Eagon, “An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology,” *Bull. Amer. Math. Soc.*, 1967.

[28] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, “A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains,” *The annals of mathematical statistics*, 1970.

[29] A. Acero, l. Deng, T. Kristjansson, and J. Zhang, “Hmm adaptation using vector taylor series for noisy speech recognition,” 2000.

[30] S. Ahadi and P. C. Woodland, “Combined bayesian and predictive techniques for rapid speaker adaptation of continuous density hidden markov models,” *Computer speech & language*, 1997.

[31] L. Bahl, P. Brown, P. de Souza, and R. Mercer, “Maximum mutual information estimation of hidden markov model parameters for speech recognition,” in

ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing, 1986.

[32] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, 1989.

[33] 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*, 2006.

[34] Y. Qin, N. Frosst, S. Sabour, C. Raffel, G. Cottrell, and G. Hinton, "Detecting and diagnosing adversarial images with class-conditional capsule reconstructions," in *International Conference on Learning Representations*, 2020.

[35] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," in *International Conference on Learning Representations*, 2018.

[36] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Joint European conference on machine learning and knowledge discovery in databases*. Springer, 2013.

[37] 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.

[38] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017.

[39] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," 2020.

[40] C. Herley and P. C. Van Oorschot, "Sok: Science, security and the elusive goal of security as a scientific pursuit," in *2017 IEEE symposium on security and privacy (SP)*, 2017.

[41] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[42] H. Kwon, H. Yoon, and K.-W. Park, "Poster: Detecting audio adversarial example through audio modification," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[43] L. Yujian and L. Bo, "A normalized levenshtein distance metric," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2007.

[44] J. Lu, T. Issaranon, and D. Forsyth, "Safetynet: Detecting and rejecting adversarial examples robustly," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[45] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[46] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, "Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions," in Proc. *ICASSP*, 2018.

[47] P. Neekhara, C. Donahue, M. Puckette, S. Dubnov, and J. McAuley, "Expediting tts synthesis with adversarial vocoding," *Proc. Interspeech*, 2019.

[48] C. Miao, S. Liang, M. Chen, J. Ma, S. Wang, and J. Xiao, "Flow-tts: A non-autoregressive network for text to speech based on flow," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2020.

[49] Bhadragiri Jagan Mohan and Ramesh Babu N., "Speech recognition using mfcc and dtw," in *2014 International Conference on Advances in Electrical Engineering (ICAEE)*, 2014.

[50] M. Ravanelli, T. Parcollet, and Y. Bengio, "The pytorch-kaldi speech recognition toolkit," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[51] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *The Journal of the Acoustical Society of America*, 1937.

[52] J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, "Fast signal reconstruction from magnitude STFT spectrogram based on spectrogram consistency," in Proc. *International Conference on Digital Audio Effects*, 2010.

[53] D. W. Griffin, Jae, S. Lim, and S. Member, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. Acoustics, Speech and Sig. Proc*, 1984.

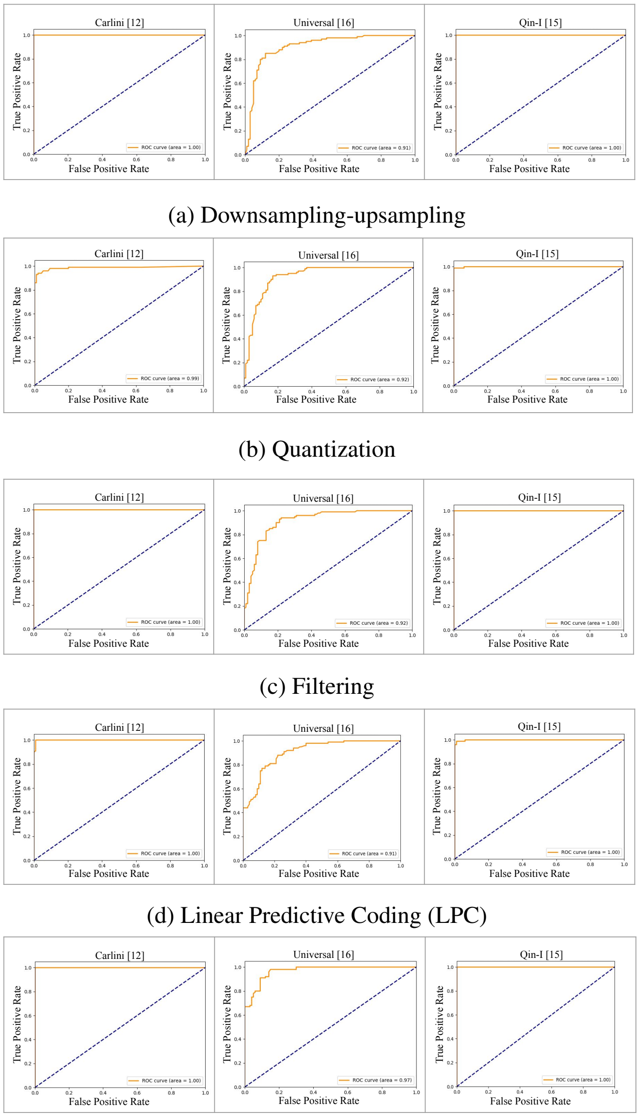
[54] Y. He, *TensorFlow implementation of Griffin-Lim algorithm*, 2017. [Online]. Available: https://github.com/candlewill/Griffin_lim

[55] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

11 Appendix

A. Receiver Operating Characteristic curves for Detection under Non-Adaptive Attacks

We provide the Receiver Operating Characteristic (ROC) curves for our detection of non-adaptive adversarial attacks using various transformation functions against three different adversarial attacks in Figure 11. The AUC scores are reported in Table 2 in Section 6.1 and included with each of the plots below. A *true positive* implies an example that is adversarial and is correctly identified as adversarial.



(a) Downsampling-upsampling

(b) Quantization

(c) Filtering

(d) Linear Predictive Coding (LPC)

(e) Mel Extraction - Inversion

Figure 11: Detection ROC curves for different transformation functions against three attacks (Carlini [11], Universal [15], Qin-I [14]) in the non-adaptive attack setting.

B. Thresholds for Detection Accuracy

Table 5 lists the detection thresholds (t) for various transformation functions for the two ASR systems studied in our work. We choose 50 original examples (separate from the first 100 used for evaluation) and construct 50 adversarial examples using each of the attack. This results in 100 adversarial-benign example pairs for DeepSpeech (constructed using Carlini [11] and Universal [15] attacks) and 100 adversarial-benign example pairs for Google Lingvo (constructed using Qin-I and Qin-R attacks [14]). Using this dataset, we obtain the threshold that achieves the best detection accuracy for each defense separately for the two ASRs. The AUC metric is threshold independent. We do not change the threshold for adaptive attack evaluation and use the same threshold as listed in Table 5.

Defense	Threshold - DeepSpeech	Threshold - Lingvo
Downsampling - Upsampling	0.48	0.48
Quantization - Dequantization	0.44	0.26
Filtering	0.32	0.31
Mel Extraction - Inversion	0.33	0.31
LPC	0.38	0.46

Table 5: Detection Threshold when using each transformation function in WaveGuard framework for DeepSpeech and Lingvo ASR systems.

C. Transfer Attacks from an Undefended Model

	Distortion metrics		*Attack Performance*		*Detection Scores*	
Defense	$\|\delta\|_\infty$	$dB_x(\delta)$	$CER(x_{adv},\tau)$	$CER(g(x_{adv}),\tau)$	AUC	Acc.
LPC	1000	-23.5	0.0	0.80	0.99	98.5%
LPC	2000	-17.4	0.0	0.83	0.99	99.0%
LPC	4000	-11.4	0.0	0.81	0.99	97.0%
LPC	8000	-5.4	0.0	0.91	0.99	99.0%
Mel Ext - Inv	1000	-23.5	0.0	0.81	0.99	98.5%
Mel Ext - Inv	2000	-17.4	0.0	0.88	0.99	97.5%
Mel Ext - Inv	4000	-11.4	0.0	0.89	0.99	98.0%
Mel Ext - Inv	8000	-5.4	0.0	0.92	0.99	98.5%

Table 6: Evaluation of Mel Extraction - Inversion and LPC transform defense against perturbations targeting an undefended DeepSpeech ASR model at different levels of magnitude.

We additionally evaluate the robustness of Mel extraction-inversion and LPC transformations against transfer attacks from an undefended model. We craft targeted adversarial examples using [11] for DeepSpeech ASR at different perturbation levels by linearly scaling the perturbation to have the desired L_∞ norm. Table 6 shows the evaluations of transfer attack at different perturbation levels. We find that attacks targeting undefended models do not break the defense using these two transformation functions even at high perturbation levels. This is because the transcription of $g(x_{adv})$ is significantly different from the target transcription and transcription of x_{adv} even at high perturbation levels thereby allowing our detector to consistently detect the adversarial samples.

D. Straight-through Gradient Estimator for LPC

We find that the LPC transform cannot be broken in an adaptive attack scenario using BPDA attack with a straight-through gradient estimator (i.e assuming identity function as the gradient of transformation function g during the backward pass). In our experiments, we started with an initial ϵ_∞ of 2000, and increased the initial distortion bound to 16000 but did not observe any improvement in the attack performance as the detector was still able to identify adversarial audio with 100% accuracy. Therefore, using our BPDA attack algorithm, we do not arrive at a solution in which both x and $g(x)$ transcribe to the target phrase even with a high amount of allowed distortion. This motivated us to design stronger adaptive attacks with differentiable LPC (Section 7.1) to find distortion bounds over which LPC transforms are not able to reliably detect adversarial examples.

	Distortion metrics			*Attack Performance*		*Detection Scores*	
Defense	ϵ_∞	$\lvert\delta\rvert_\infty$	$dB_x(\delta)$	$CER(x_{adv},\tau)$	$CER(g(x_{adv}),\tau)$	AUC	Acc.
LPC	2000	2000	-15.9	0.31	0.85	1.0	100%
LPC	16000	16000	2.1	0.34	0.85	1.0	100%

Table 7: Evaluation of LPC transform against straight-through gradient estimator.