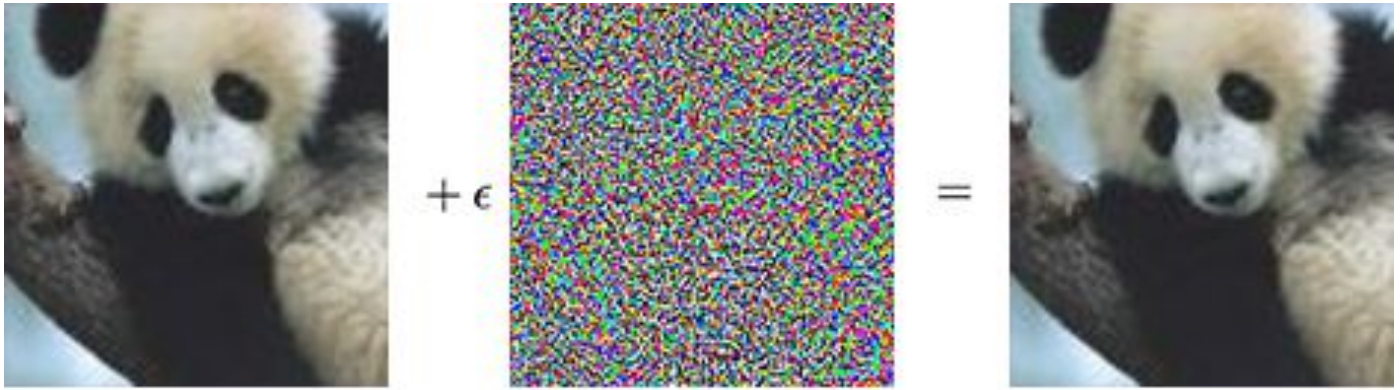
Waveguard: Understanding and Mitigating Audio Adversarial Examples

Hussain, S., Neekhara, P., Dubnov, S., McAuley, J., & Koushanfar, F.

Usenix 21’

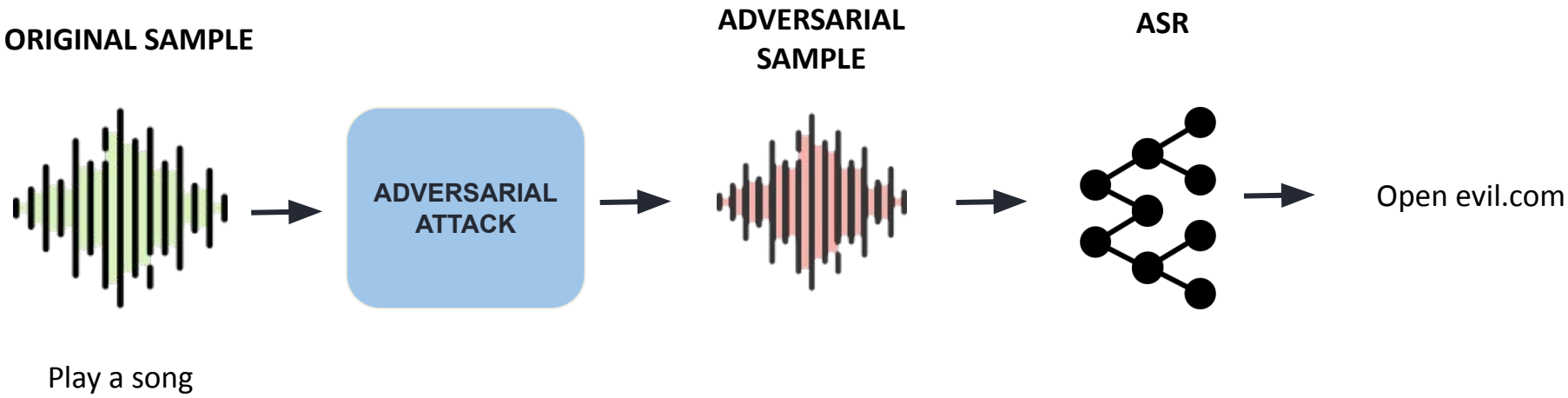
Attacks on Deep learning with adversarial samples



"panda"
57.7% confidence

"gibbon"
99.3% confidence

Adversarial audio attacks

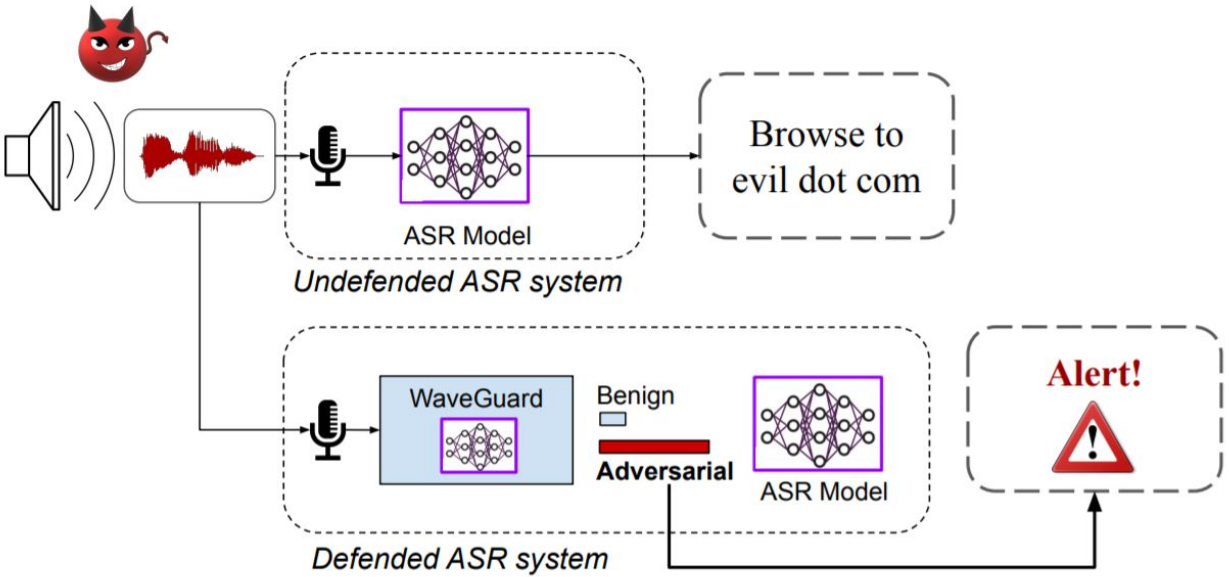


Paper in a Nutshell

Investigates defense mechanisms on different audio adversarial example types

- ➔ Evaluate lossy transformations to detect state-of the art target/untargeted attacks
- ➔ Evaluate defenses against adaptive attack (white box)
- ➔ Evaluate transformations overhead

WAVEGUARD



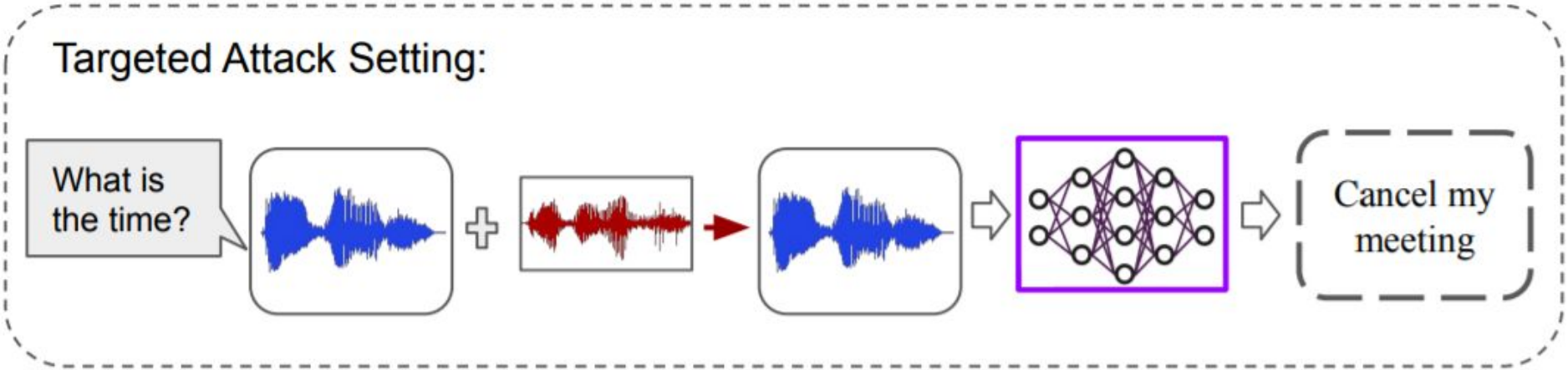
1. Detects adversarial samples
2. Alerts ASR system

Threat models

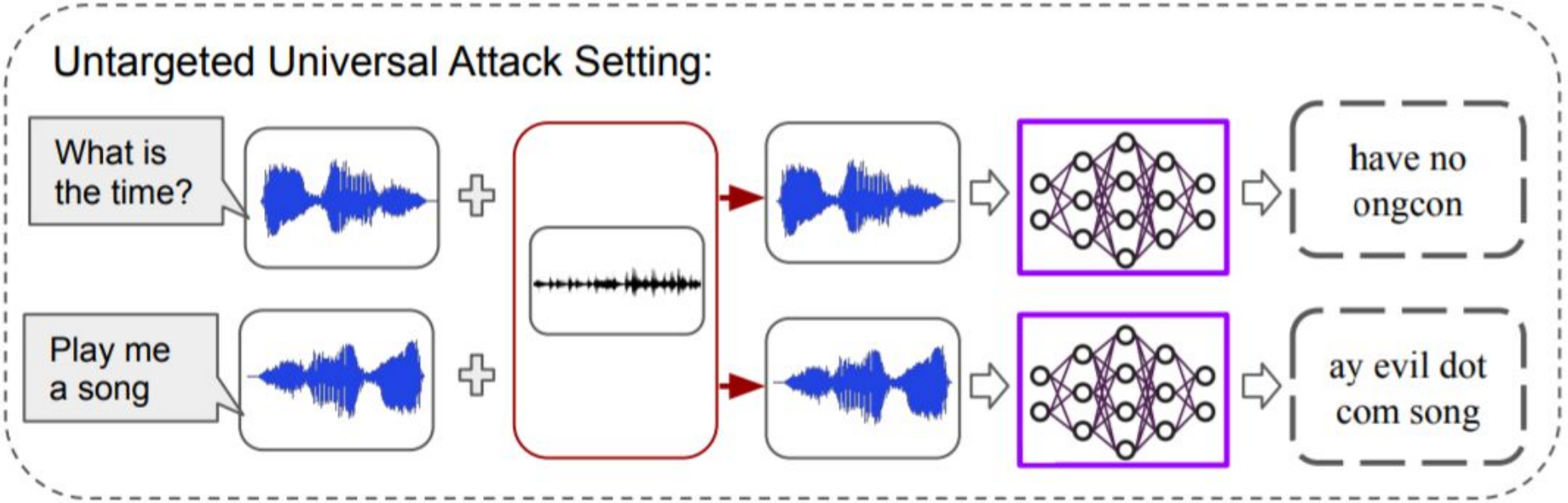
Paper considers 2 scenarios:

1) Targeted attacks (Target transcription)
2) Untargeted attacks (Disrupt the ASR)
3) Adaptive attacks (white box access to defense)

Threat models



Threat models



WAVEGUARD

- Performs transformations on incoming audio samples
- Measure difference in the normalized edit distance between the original audio transcription and the transcription after the transformations

Assumption: adversarial attacks are unstable and small changes can affect the prediction significantly.

WAVEGUARD

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.

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

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

WAVEGUARD

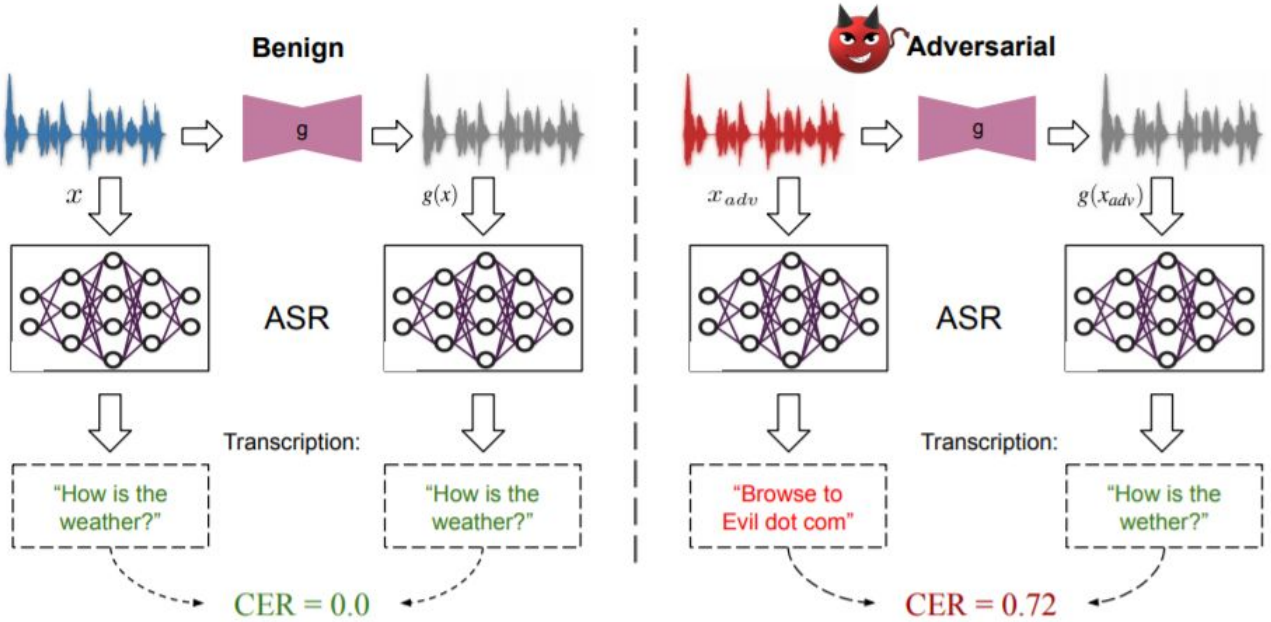


Figure 3: WaveGuard Defense Framework: We first processes the input audio x using an audio transformation function g to obtain $g(x)$. Then the ASR transcriptions or x and $g(x)$ are compared. An input is classified as *adversarial* if the difference between the transcriptions of x and $g(x)$ exceeds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

WAVEGUARD

Build Waveguard with transformations:

- Quantization-Dequantization
- Down-sampling and Up-sampling
- Filtering
- Mel Spectrogram Extraction and Inversion
- Linear Predictive Coding

Quantization-Dequantization

- Waveform sample is quantized to lower bits precision
- It is then scaled back to floating point to approximate original data

Down-sampling and Up-sampling

- Down-sample the original waveform (16 kHz in the paper), to a lower sampling rate
- Upsample by estimating the waveform at its original sampling rate using interpolation

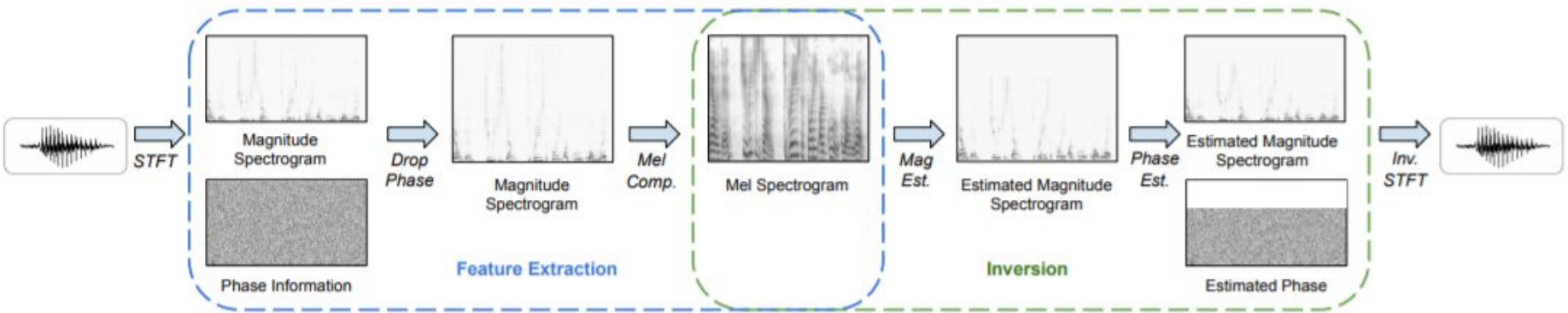
Filtering

Modified low-pass and high-pass filter, shelf filters

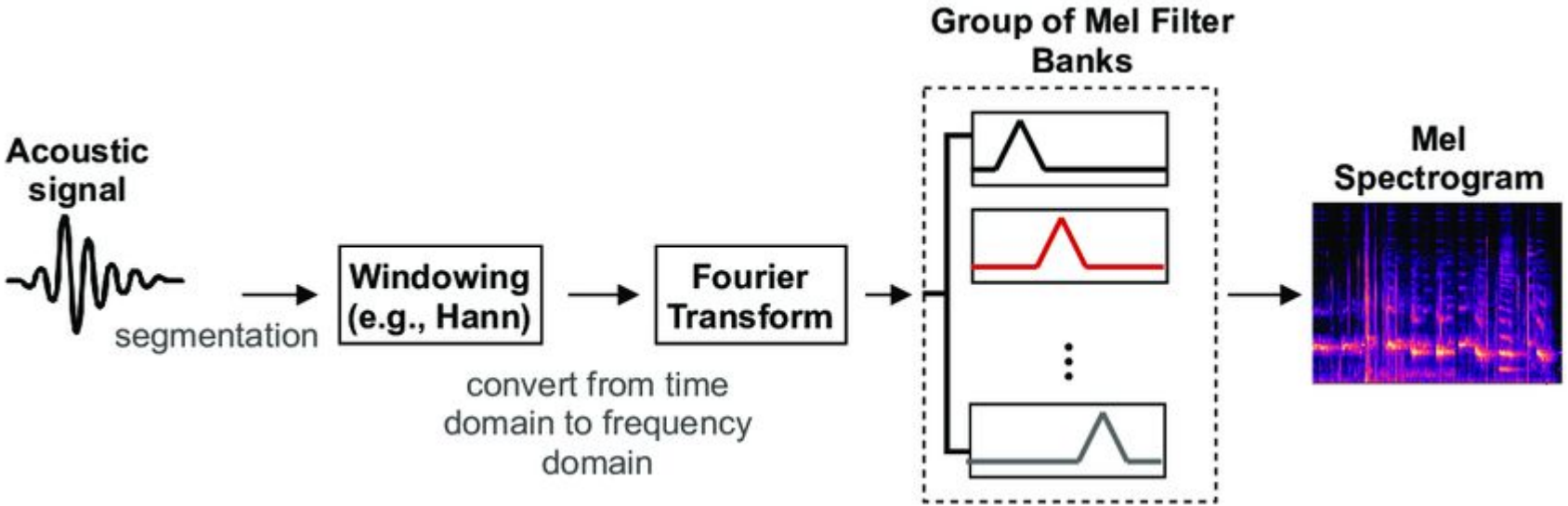
- Low/high pass : filters high and low frequencies
- Low/high **Shelf filters** : reduce amplitude of high/low frequencies

Mel Spectrogram Extraction and Inversion

Compute Mel Spectrogram and revert back to waveform



Mel Spectrogram Extraction and Inversion



Linear Predictive Coding

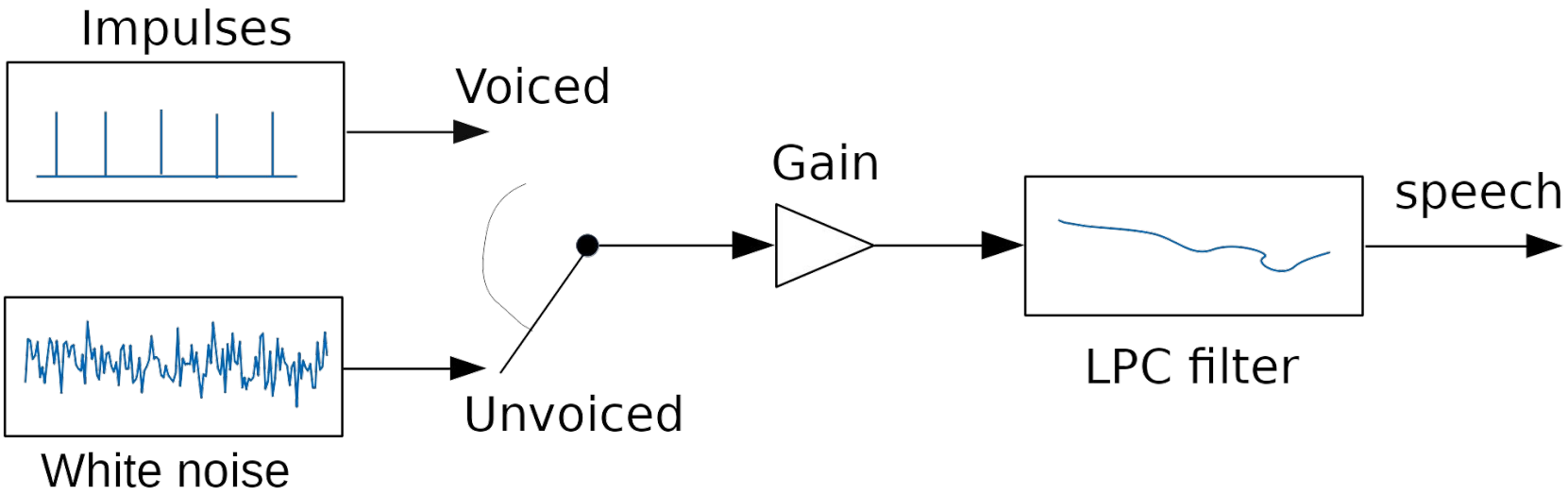
LP is a speech coding / feature extraction technique

- Speech can be modeled as the output of a linear, time-varying system, excited by either quasi-periodic pulses or noise

- Each speech sample can be closely approximated as a linear combination of past samples

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

Linear Predictive Coding



Experiments

- **Q1:** How effective are the transformations against the non-adaptive SOTA attacks?
- **Q2:** How about and adaptive attack?
- **Q3:** How efficient is it ?

RQ1: Non-adaptive attacks

Experiments setup

- **Dataset: Mozilla Common voice dataset**
  - 582 hours of audio, 400,000 samples (english)
  - Test set : 100 samples
  - For each attack, create 100 adversarial samples
- **Non-adaptive Attacks:**
  - Carlini
  - Qin-I
  - Qin- R
  - Universal

Attack: Carlini

Audio adversarial examples: Targeted attacks on speech-to-text (Carlini et al. , 2018)

➔ Minimizes the Connectionist Temporal Classification (CTC) loss between the target transcription and the ASR's prediction

- Targeted
- White box
- Targets Mozilla Deepspeech

Attack: Qin-I

Imperceptible, robust, and targeted adversarial examples for automatic speech recognition (Qin et al., 2019)

➔ Based on psycho-acoustic hiding

- Targeted
- White box
- Targets Google Lingvo

Attack: Qin-R

Imperceptible, robust, and targeted adversarial examples for automatic speech recognition (Qin et al., 2019)

➔ Incorporates noise simulation during training of the adversarial perturbation which simulate room environments

- More robust to being played over-the-air
- White box
- Targets Google Lingvo

Attack: Universal

Universal adversarial perturbations for speech recognition systems (Neekhara et al., 2019)

➔ **Finds perturbation that disrupts ASR transcription**

- Independent of input audio
- White box
- Targets Mozilla DeepSpeech

Attacks Evaluation

Carlini : 100 % success rate

Qin-I : 100% success rate

Qin-R: 47% success rate

Universal: 81% success rate

Detection AUC (Carlini)

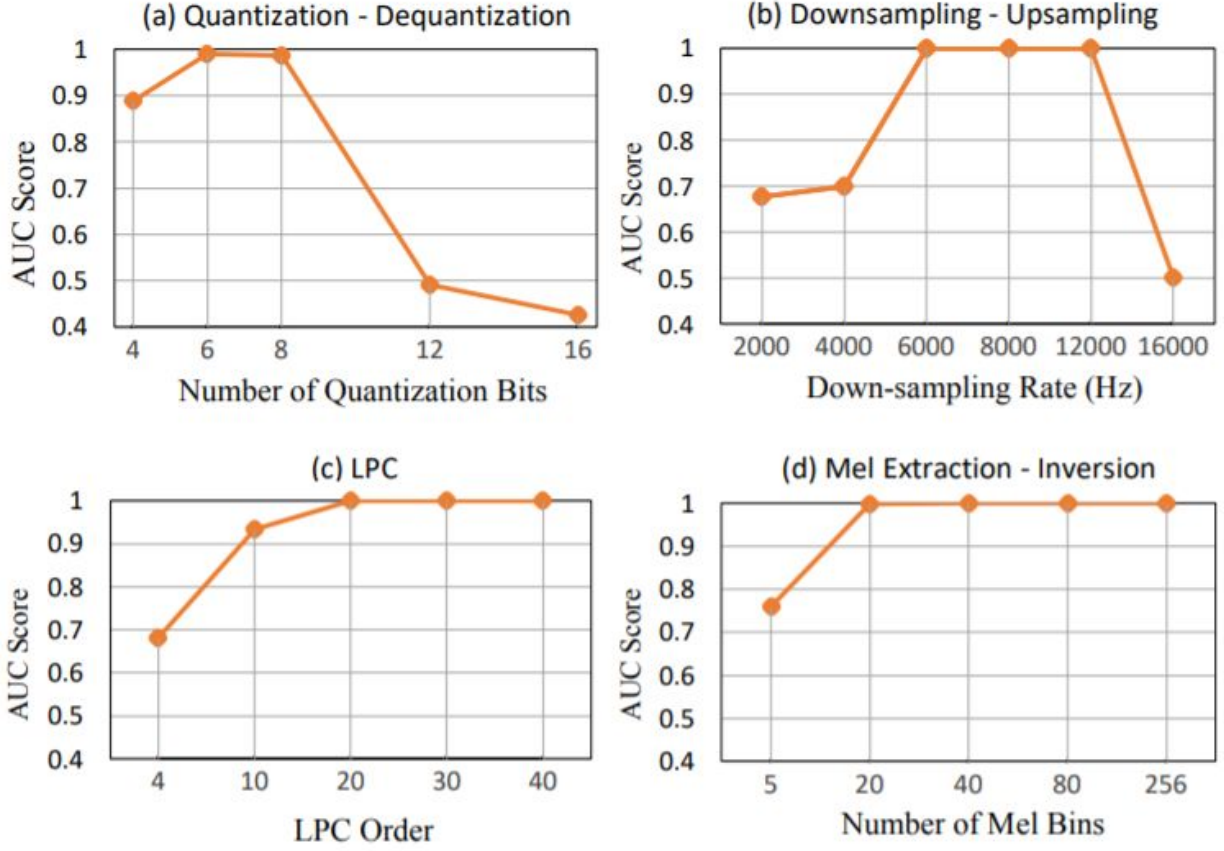


Figure 6: Detection AUC Scores against Carlini attack at varying compression levels for the following transforms: (a) Quantization - Dequantization; (b) Downsampling - Upsampling; (c) Linear Predictive Coding (LPC); and (d) Mel Spectrogram Extraction- Inversion

Detection Accuracy

	AUC Score				*Detection Accuracy*			
Defense	Carlini	Universal	Qin-I	Qin-R	Carlini	Universal	Qin-I	Qin-R
Downsampling - Upsampling	1.00	0.91	1.00	1.00	100%	88%	100%	100%
Quantization - Dequantization	0.99	0.92	1.00	0.93	98.5%	88%	99%	95%
Filtering	1.00	0.92	1.00	1.00	99.5%	86%	100%	100%
Mel Extraction - Inversion	1.00	0.97	1.00	1.00	100%	92%	100%	100%
LPC	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).

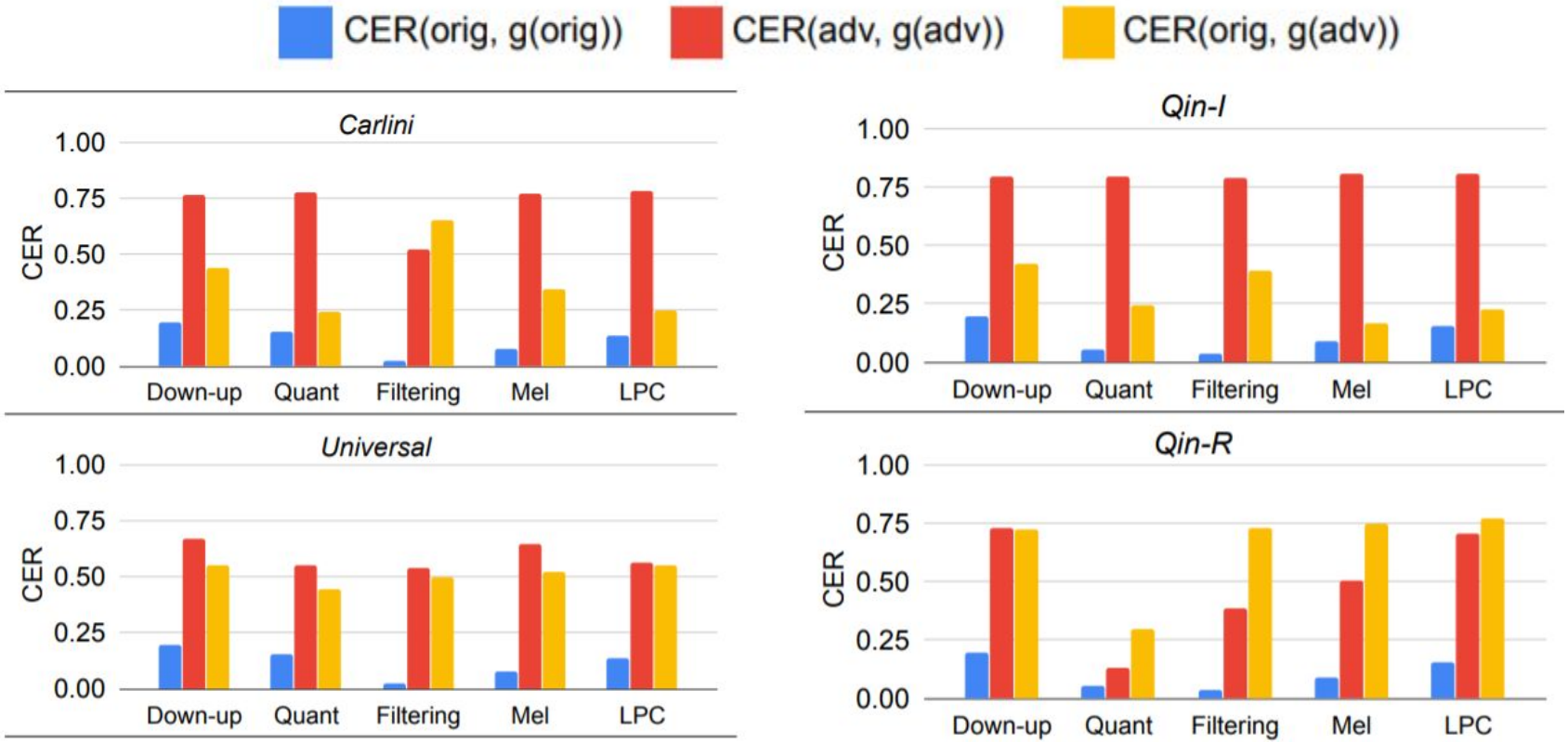
Effect on the ASR transcription

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"

Effect on the ASR transcription

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"

Effect on the ASR transcription



RQ2: Adaptive attack

Adaptive Attack

Carlini attack that minimizes $d(C(x), C(g(x)))$

- Targeted
- White box

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

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

Use Backward Pass Differentiable Approximation to backpropagate over transformations

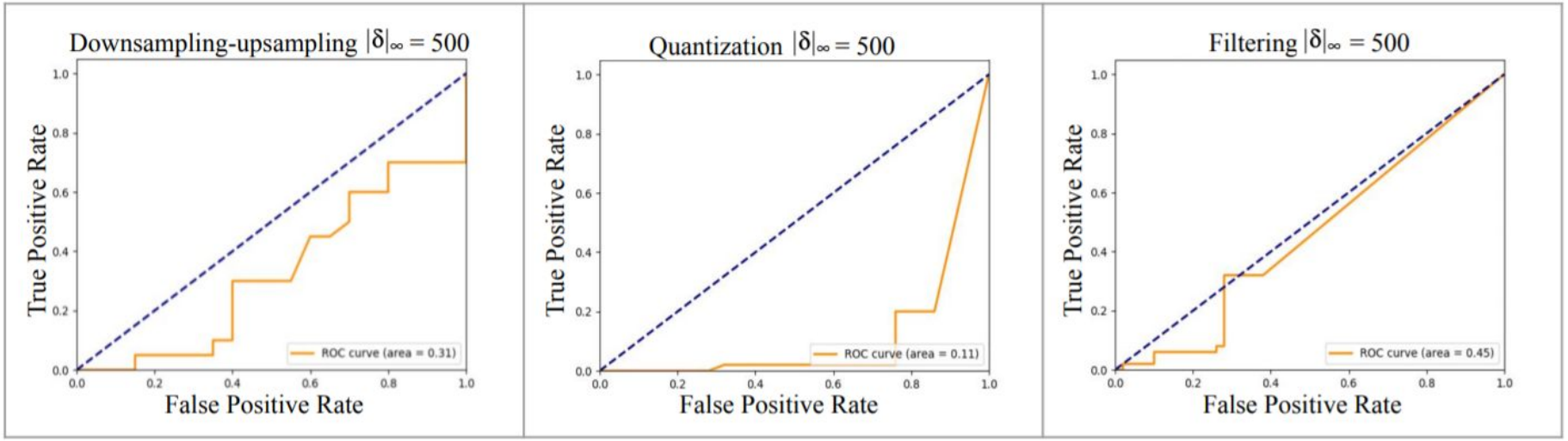
Adaptive Attack: Evaluation

1) How efficient is the attack?
2) How effective is the defense at detecting this attack?

Metrics to measure imperceptibility :

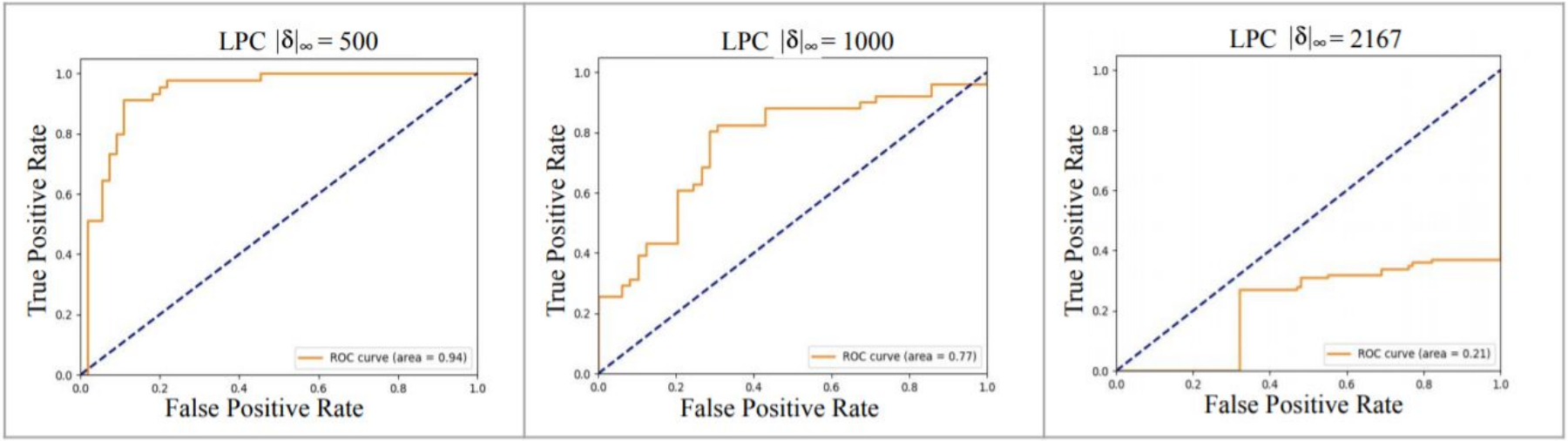
- Distortion
- Loudness in Decibels (dB)

Detection ROC curves



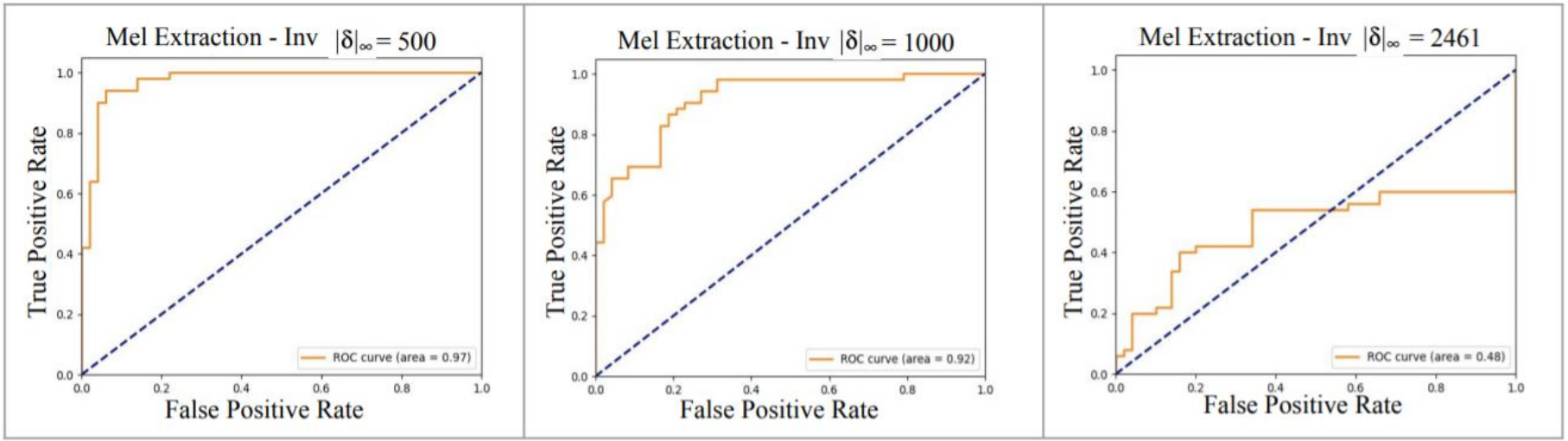
(a) Downsampling-upsampling, Quantization and Filtering

Detection ROC curves



(b) Linear Predictive Coding (LPC)

Detection ROC curves



(c) Mel Extraction - Inversion

	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%

Effect on the ASR transcription

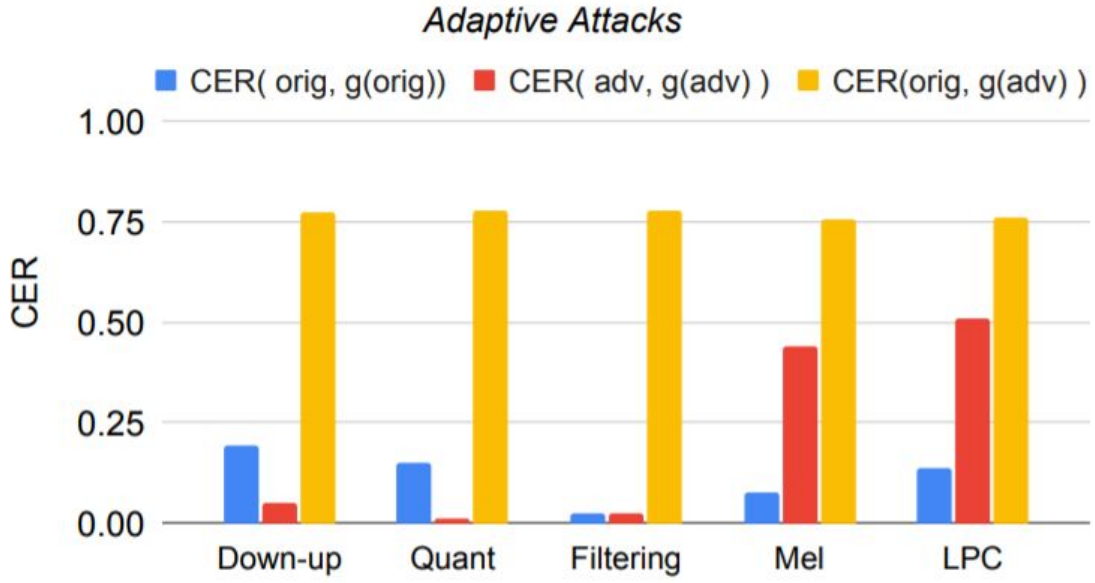


Figure 10: Mean CER between the ASR transcriptions of un-transformed (x) and transformed $(g(x))$ audio for adaptive attacks with an initial distortion $\varepsilon_\infty = 500$.

RQ3: Waveguard overhead

RQ3: Waveguard overhead

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.

Discussion & Limitations

1. LPC and MEFL frequency work best against attacks and og transcription retrieval
2. SOTA non-adaptive attacks are easily detectable
3. This approach does not attempt to retrieve the original signal, but some transformations are better than others, depending on the attack
4. Relatively little overhead
5. Selected transformations are not effective against adaptive attacks

Conclusion

1. Waveguard is a system to detect malicious audio samples by comparing the original audio and the audio passed through several transformations.
2. This paper studies the impact of different functions on SOTA attacks
3. Waveguard achieves 100% detection of non-adaptive attacks (targeted/untargeted and white box/black box)
4. Speech feature extraction methods are the most promising
5. Adaptive attacks are harder to detect and to retrieve the original transcription

Relevant Papers

1. **Defenses**
    a. **Characterizing audio adversarial examples using temporal dependency,** Yang, Zhuolin, et al., 2018
    b. **A Unified Framework for Detecting Audio Adversarial Examples**, Du, Xia, Chi-Man Pun, and Zheng Zhang, 2020.
2. **Attacks:**
    a. **AdvPulse: Universal, Synchronization-free, and Targeted Audio Adversarial Attacks via Subsecond Perturbations**, Li, Zhuohang, et al, 2020

Thank you!