Robust Detection of Machine-induced Audio Attacks in Intelligent Audio Systems with Microphone Array

Zhuohang Li
University of Tennessee
Knoxville, TN, USA
zli96@vols.utk.edu

Cong Shi
Rutgers University
New Brunswick, NJ, USA
cs1421@winlab.rutgers.edu

Tianfang Zhang
Rutgers University
New Brunswick, NJ, USA
tz203@scarletmail.rutgers.edu

Yi Xie
Rutgers University
New Brunswick, NJ, USA
yx238@scarletmail.rutgers.edu

Jian Liu*
University of Tennessee
Knoxville, TN, USA
jliu@utk.edu

Bo Yuan
Rutgers University
New Brunswick, NJ, USA
bo.yuan@soe.rutgers.edu

Yingying Chen
Rutgers University
New Brunswick, NJ, USA
yingche@scarletmail.rutgers.edu

ABSTRACT

With the popularity of intelligent audio systems in recent years, their vulnerabilities have become an increasing public concern. Existing studies have designed a set of machine-induced audio attacks[1], such as replay attacks, synthesis attacks, hidden voice commands, inaudible attacks, and audio adversarial examples, which could expose users to serious security and privacy threats. To defend against these attacks, existing efforts have been treating them individually. While they have yielded reasonably good performance in certain cases, they can hardly be combined into an all-in-one solution to be deployed on the audio systems in practice. Additionally, modern intelligent audio devices, such as Amazon Echo and Apple HomePod, usually come equipped with microphone arrays for far-field voice recognition and noise reduction. Existing defense strategies have been focusing on single- and dual-channel audio, while only few studies have explored using multi-channel microphone array for defending specific types of audio attack. Motivated by the lack of systematic research on defending miscellaneous audio attacks and the potential benefits of multi-channel audio, this paper builds a holistic solution for detecting machine-induced audio attacks leveraging multi-channel microphone arrays on modern intelligent audio systems. Specifically, we utilize magnitude and phase spectrograms of multi-channel audio to extract spatial information and leverage a deep learning model to detect the fundamental difference between human speech and adversarial audio generated by the playback machines. Moreover, we adopt an unsupervised domain adaptation training framework to further improve the model's generalizability in new acoustic environments. Evaluation is conducted under various settings on a public multi-channel replay attack dataset and a self-collected multi-channel audio attack dataset involving 5 types of advanced audio attacks. The results show that our method can achieve an equal error rate (EER) as low as 6.6% in detecting a variety of machine-induced attacks. Even in new acoustic environments, our method can still achieve an EER as low as 8.8%.

CCS CONCEPTS

• **Security and privacy** → *Systems security*;

KEYWORDS

intelligent audio system; audio attack; microphone array

ACM Reference Format:
Zhuohang Li, Cong Shi, Tianfang Zhang, Yi Xie, Jian Liu, Bo Yuan, and Yingying Chen. 2021. Robust Detection of Machine-induced Audio Attacks in Intelligent Audio Systems with Microphone Array. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security (CCS '21), November 15–19, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3460120.3484755

1 INTRODUCTION

During the past decade, the adoption of intelligent audio systems has surged in both residential and industrial sectors, as they provide a convenient interface for users to control and interact with smart devices through voice commands. In particular, voice assistants such as Amazon Alexa, Google Assistant, and Apple Siri have been integrated into various platforms, enabling users to conveniently control different aspects of their daily lives, such as smart home appliance controls, online purchases, personal schedule/memo inquiries, and smart vehicle operations, etc.

With such widespread applications, the vulnerabilities of these intelligent audio systems to various types of audio attacks have become a rising security concern. For instance, *replay attack* [32, 74], which attempts to bypass the authentication process simply using a recording from the victim, has long been one of the dominant sources of audio spoofing attacks. *Synthesis attack* [46, 78] utilizing text-to-speech engines to mimic the victim's voice is a common alternative when the victim's speech sample cannot be directly obtained. Besides these conventional attack approaches, recent studies have revealed new vulnerabilities including *hidden voice commands* [7, 13, 68], *inaudible attacks* [85], and *audio adversarial examples* [14, 81], which exploit either the gap between machine

*Corresponding author.

[1]"Machine-induced attack" refers to the audio attack that requires to use a playback device (e.g., loudspeaker or ultrasound speaker) to play the crafted speech samples.

CCS '21, November 15–19, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8454-4/21/11…$15.00
https://doi.org/10.1145/3460120.3484755

and human perceptions or the intrinsic vulnerability that lies in deep learning models to make the attack unnoticeable.

Existing Defenses. The emerging attack vectors of intelligent audio systems demand a general defense strategy to secure the voice user interface against the disclosed suite of audio attacks. However, most existing studies treat each type of attack differently and seek to design specific mechanisms against each individual attack. The replay attack is the most studied audio attack, where many features derived from speech signals have been considered for designing classifiers to distinguish replayed speech from human speech, such as Mel Frequency Cepstral Coefficient (MFCC) [24], Constant Q Cepstral Coefficients (CQCC) [66], Linear Prediction Cepstral Coefficient (LPCC) [76], and Rectangular Frequency Cepstral Coefficient (RFCC) [24]. In addition to these power spectrum features, relative phase shift [17], pitch patterns [33], and neuron activation patterns [73], along with other spectral features [42] have been proposed to help discriminate between human and synthetic speech. Countermeasures to hidden voice commands, as pointed out in the original work [13], include training a classifier (i.e., logistic regression) with the acoustic features extracted from hidden voice commands and normal speech commands which was shown to almost fully defend against this type of attack. A recent work [71] also showed the potential of using the built-in motion sensors of smartphones to defeat hidden voice commands. To defend against inaudible attacks, microphone enhancement, baseband cancellation, or learning unique features of modulated voice commands which are distinctive from genuine ones have been considered [85]. Additionally, detection method based on propagation attenuation [65] and active defense using an emitted inaudible "guard" signal to cancel the attack [27] have also been explored. As for defending audio adversarial examples, various methods leveraging audio transformation [82] or transcription analysis [30, 83] have been proposed.

Limitation of Prior Work. As described above, existing studies have been mostly focusing on designing dedicated mechanisms for defending individual attack. These mechanisms are designed from different perspectives and require different sensing modalities or additional hardware modules, making them almost impossible to be combined and deployed onto an audio device in practice. Thus, a lightweight holistic defense strategy against all existing audio attacks is highly desirable. In addition, most off-the-shelf intelligent audio systems are equipped with a microphone array for far-field voice recognition, noise reduction, and acoustic echo cancellation [8]. For instance, Amazon Echo (4th generation) has 6 mics; Amazon Echo Auto has 8 mics; Apple HomePod has 6 mics, etc[2]. In contrast, most existing efforts for detecting replayed audio are based on single-channel recordings. Several studies [49, 64, 80, 87] go beyond a single channel to explore dual-channel stereo recordings on smartphones. However, this line of work often suffers from short detection range and still fails to exploit the rich sensing capability of multi-channel microphone arrays that are ubiquitous in intelligent audio systems. As one of the few works that exploit multi-channel audio, EarArray [65] proposes to utilize the estimated attenuation rate of the ultrasound signal via microphone array for detecting inaudible attacks but does not generalize to other audio attacks. A more recent work by Gong *et al.* [23] demonstrates that compared to single-channel audio, multi-channel audio can help improve the performance (up to 34.9%) for detecting replay attacks. However, this work leverages beamforming technique [10] to filter and combine multi-channel audio signals into a single-channel signal, which loses distinct spatial information carried on other channels. In addition, this work merely focuses on replay attacks and does not address other more advanced audio attacks.

[2]The number of microphones on mainstream intelligent audio devices is summarized in Appendix Table 8.

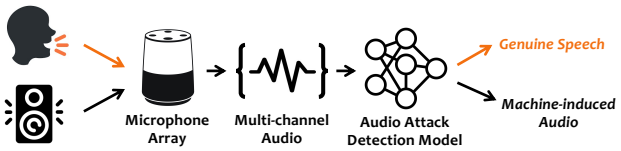


Figure 1: Overview of the proposed approach.

Benefits of Leveraging Multi-channel Audio. Compared to existing single-channel-based solutions, multi-channel audio attack detection offers benefits in several aspects: 1. *Spatial Feature*: In addition to the temporal and spectral information used for single-channel audio attack detection, multi-channel audio also encodes important spatial information (e.g., angle of arrival (AoA) and time difference of arrival (TDoA)) that is harder for the attacker to manipulate; 2. *Detection Range*: Existing single-channel or dual-channel audio attack detection systems (e.g., [49, 86, 87]) only perform well in close-range scenarios (e.g., talking on phones) as its performance decreases drastically when the microphone is far from the user, while multi-channel audio signals collected from microphone arrays can be utilized to achieve sound source localization and speech enhancement by reducing noise and reverberations which are critical for far-field detection; 3. *Device Compatibility*: To achieve far-field hands-free voice control, intelligent audio devices (e.g., Amazon Echo and Apple HomePod) often come equipped with a multi-channel microphone array. Thus multi-channel audio attack detection solutions can be directly deployed to these devices to obtain enhanced performance without requiring extra hardware.

Proposed Work. Motivated by the potential benefits of multi-channel audio, in this work, we develop a holistic solution to detect various machine-induced audio attacks leveraging multi-channel microphone arrays that are available on intelligent audio systems. As shown in Figure 1, our method draws inspiration from the observation that all audio attacks require a playback device (e.g., loudspeaker or ultrasound speaker) to play the crafted attack speech sample, while genuine speech is uttered from human vocal cords. This inherent difference of sound production will be carried over into the produced audio, resulting in different patterns in signal frequency and directivity [22], which can be captured by microphone arrays and further utilized to differentiate genuine speech from machine-induced attack audio. Instead of manually searching for the optimal set of features, we resort to a learning-based approach where the model can automatically adapt to any attack method, microphone configuration, or acoustic environment that is represented in the training dataset, without requiring to be explicitly tuned. Moreover, different from a conventional approach [23] that utilizes beamforming to filter and combine multi-channel audio signal into one single-channel signal, we make use of audio signals from all available channels separately by forming 3D feature maps

with magnitude-phase spectrograms so that the important spatial information is preserved throughout the whole process. In addition, to enable efficient detection in unseen environments, we exploit unsupervised domain adaptation training to help the learned model adapt to new acoustic environments without requiring labeled data. We also explore different model configurations to design a compact model that suits mobile applications without sacrificing much detection accuracy. We summarize our main contributions as follows:

- We dissect existing machine-induced audio attacks, including replay attacks, synthesis attacks, hidden voice commands, inaudible attacks, and audio adversarial examples, and design a holistic defense strategy leveraging multi-channel audio recorded by the microphone array equipped on intelligent audio systems.
- We build a deep learning model leveraging both magnitude and phase information derived from multi-channel audio to achieve accurate and robust detection of audio attacks without handcrafted features. Moreover, we adopt the unsupervised domain adaptation framework to achieve environment-independent detection, making the system still work well when deployed in a new environment.
- To evaluate our proposed holistic solution, we re-implement a set of representative advanced audio attacks and collect a dataset of voice recordings of the reproduced adversarial speech samples in different environmental conditions with various playback/recording devices.
- Extensive experiments on a public multi-channel replay attack dataset and an empirically-collected advanced audio attack dataset showed that our method can achieve up to 6.6% equal error rate (EER) in detecting these machine-induced audio attacks. Even in a new environment, our environment-independent solution can still achieve reasonably good performance.

2 RELATED WORK

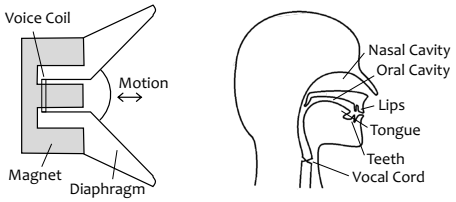
2.1 Machine-induced Audio Attacks

Due to the open nature of voice access, intelligent audio systems have been proven to be vulnerable to many spoofing attacks, such as conventional replay attacks [32, 76], synthesis attacks [39, 46, 75, 78] and some other more advanced audio attacks leveraging modulated attacking sound (e.g., hidden voice commands [7, 13, 68]). Among these attacks, replay attacks are the most accessible to the adversary since it simply involves recording a victim's speech samples with a handy recording device (e.g., a smartphone) and replaying the speech samples for the attack. A recent study [74] also designed modulated replay attacks that align the frequency domain distortions induced in the replay process, rendering the replayed sound more similar to genuine human speech. When collecting speech samples is difficult, the adversary can also launch synthesis attacks that produce speech samples mimicking the victim's voice characteristics (e.g., pitch range, frequency component distributions). These attacks usually leverage voice synthesis models [46, 75] to convert text into the target speech of a victim, by using only a small number of the victim's voice samples for training (e.g., collected through the Internet or public speech). In addition, the adversary can modify the voice samples from arbitrary speakers to make them sound like the victim's voice for the attack [39, 78]. Due to the recent advancements in deep learning, such speech synthesis models can produce natural-sounding speech, making the attacks difficult to be detected.

In addition to these conventional attacks through replaying human-sounding speech signals, recent studies demonstrated the potential of generating unintelligible or even inaudible attacking sound, by leveraging the perception gap between humans and machines [7, 13, 41, 68, 85] or the intrinsic vulnerability of embedded deep learning models [14, 35, 36, 41, 84]. For example, hidden voice commands [13, 68] convert speech into obfuscated voice commands that are recognizable to speech recognition models while remaining unintelligible to humans. As another example, inaudible attacks [41, 85] modulate the recorded speech samples onto ultrasonic frequency bands (e.g., over 20kHz), which are completely inaudible to human listeners but can be demodulated by the microphones due to their non-linearity properties. Furthermore, as current embedded speech/speaker recognition engines are mostly based on deep neural network (DNN) models, the adversary can explore the models' inherent vulnerabilities to generate well-crafted adversarial perturbations to access intelligent audio systems. Such attacks either add imperceptible perturbations to replayed audio [14, 35] or embed speech samples into ambient noises/background music [84] to spoof the speech/speaker recognition engines, making the model yield adversary-desired output (e.g., speaker identity or speech content). A more recent study [36] even developed practical adversarial examples that injects adversarial perturbations onto streaming audio inputs (e.g., live human speech) in an unsynchronized manner, demonstrating a severe threat to intelligent audio systems.

2.2 Existing Attack Detection Strategies

Although significant research efforts have been devoted to developing attack detection methods to secure voice access, few studies have investigated using readily available microphone arrays on modern intelligent audio systems to further enhance their security levels. Most existing studies rely on extracting frequency domain features from single-channel audio to differentiate replayed or synthesized voice from genuine human speech. For instance, power spectrum features [24, 33, 42], relative phase shifts [17], and magnetic field distortions [15] are exploited to detect replayed and synthesized speech. A recent study, VOID [9], leveraged the spectral features extracted from single-channel audio to detect various audio attacks. However, the lack of using multi-channel audio and the spatial information makes it still vulnerable to many advanced attacks, such as modulated replay attacks [74], which aligns the frequency domain distortions induced in the replay process. Furthermore, several studies performed liveness detection using dynamic acoustic features from dual-channel audio, such as pop noises from breaths [49], cross-correlation of stereo signal [80], and time-difference-of-arrival changes of phoneme sounds [87]. However, such dynamic acoustic features only exist in proximity and require the microphones to be placed close to the user's mouth (e.g., when talking to a smartphone). Gong *et al.* [23] demonstrated the potential of using multi-channel audio to defend replay attack, which shows a significant improvement compared to using single-channel audio (up to 34.9%), but the proposed system can only address replay attacks. More importantly, this work treats multiple audio channels as a whole and combines multi-channel audio



(a) Loudspeaker (b) Human

Figure 2: Comparison of sound production mechanisms.

signals into a single-channel signal, which loses distinct spatial information among the channels.

To mitigate the threats introduced by more advanced audio attacks, recent research studies developed various approaches to detect the unintelligible and inaudible attacking signals [13, 27, 36, 71, 83, 85]. To detect hidden voice commands, Carlini *et al.* [13] exploited a classifier (i.e., logistic regression) trained with frequency domain features (e.g., MFCCs, spectral entropy). Wang *et al.* [71] proposed converting the audio recordings into vibration signals (captured via motion sensors) to reveal the unique spectral characteristics of hidden voice commands. Regarding inaudible attacks, researchers have explored using frequency domain analysis [85] to detect the ultrasound signals or emitting inaudible "guard" signals to cancel the impacts of the attack [27]. More recently, to defeat inaudible voice commands (known as DolphinAttack [85]), *EarArray* [65] proposes to utilize the estimated attenuation rate via microphone array to differentiate ultrasound sounds from audible sounds. Furthermore, signal filtering, quantization, audio compression, downsampling, and adversarial training have shown to be effective to defend against audio adversarial examples [36, 83]. Although aforementioned studies, using either software-based approaches or dedicated hardware, show reasonably good performance in defending against individual attacks, it is almost impossible to combine them together as an all-in-one solution for practical deployment.

Furthermore, some researchers proposed using extra devices including smart glasses [19], smartphone [11], wearable [48], or headphone [21], to capture additional voice characteristics to perform user authentication. These investigations leveraged either unique vibration patterns (e.g., body-surface vibrations [19], air-borne vibrations [48]) or the direction of speech (e.g., angle of arrival [11]) to confirm the authenticity of the sound source. However, these approaches require additional devices, which could add extra cost and are not always applicable in practice. *CaField* [64] achieves continuous speaker verification by leveraging two on-board microphones of a smartphone to capture the acoustic features embedded in sound fields during propagation. Despite its improved usability, this method requires the smartphone to be held at a relatively close distance to the user's mouth and the holding posture/position needs to be consistent across continuous verification sessions. Thus, it is not suitable for the broader context of intelligent audio systems such as smart speakers.

Different from existing approaches, we develop a holistic defense system by leveraging multi-channel microphone arrays that are readily available in modern intelligent audio devices. Relying on both temporal and spatial information extracted from multi-channel audio, our system can detect a variety of existing machine-induced voice attacks through holistic training.

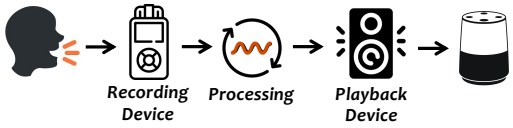


Figure 3: Typical process of audio attacks.

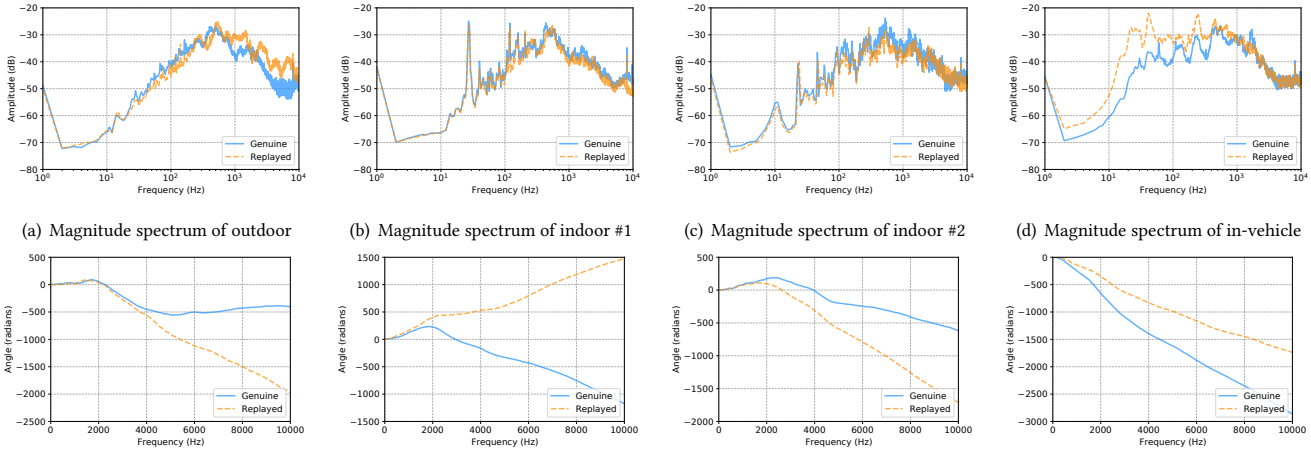
3 MULTI-CHANNEL AUDIO ANALYSIS

In this section, we explore potential acoustic features to differentiate machine-induced audio from human speech as well as validating the benefits of leveraging multi-channel audio through thoroughly analyzing a public multi-channel replay audio dataset, ReMASC [22].

3.1 Characteristics of Machine-induced Audio

Machine vs. Human Production of Sound. Machine (i.e., loudspeakers) produces sound by moving a diaphragm back and forth along one dimension to emit sound waves. As shown in Figure 2(a), during sound production, an electric current flows through the vocal coil, inducing a magnetic field that interacts with the permanent magnet and creates a force that drives the diaphragm, causing it to vibrate. Differently, as depicted in Figure 2(b), human voice production involves multiple physiological components including lungs, vocal cords, and vocal tract, and can be generally viewed as a two-stage process where a raw sound is first produced by a source and then shaped in the vocal tract [53]. Specifically, there are three different sources of speech sounds. The first type of source is vocal cord vibration, which is produced during the phonation of voiced sounds: the air stream generated by lungs flows through an open vocal tract and sets the vocal cords to oscillate, creating vowel sounds such as [a], [e], [i] and [o]. The second source of speech sound is air turbulence, which is generated by constricting the vocal tract with teeth, tongue, or lips to produce high-velocity airflow. The noises generated by the air is then shaped by the vocal tract to form consonant sounds such as [f], [s], [v] and [z]. The third source is created by completely blocking the airflow toward the front of the mouth and then followed by the sudden release of the air, which results in plosive consonants such as [k], [p] and [t]. Compared to machine-induced sound, human speech is produced from different locations within the vocal tract (e.g., oral and nasal cavities) and further shaped by the resonances of the vocal tract system. These differences result in traceable patterns in spectral energy distribution [12, 76] and propagation path [87], which will all be reflected in the magnitude and phase domain features.

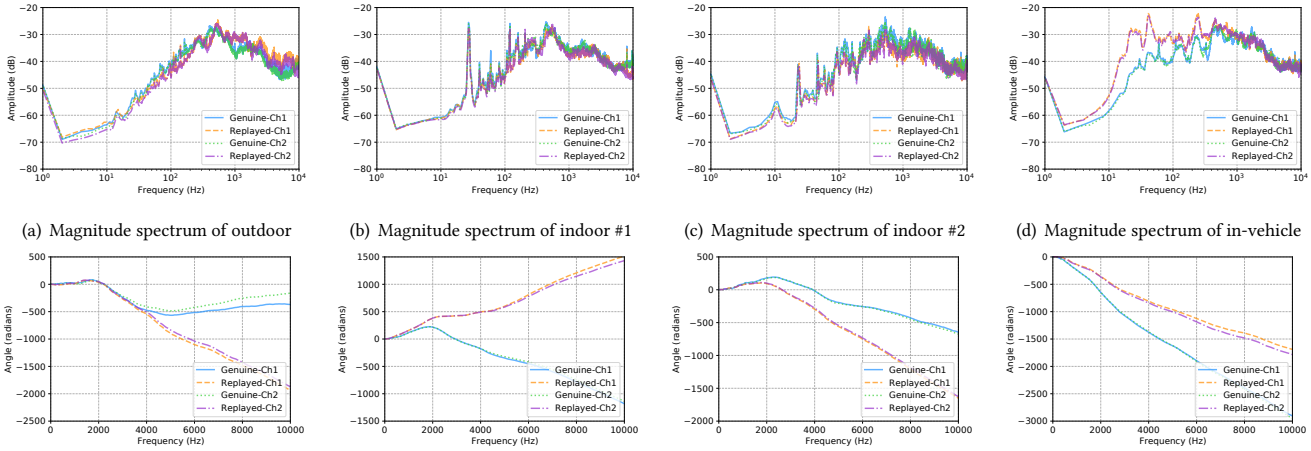
Audio Attack Process. Figure 3 illustrates the typical process of machine-induced audio attacks. The attacker first records speech commands using a recording device, then plays the recorded audio using a playback device when launching the attack. For some advanced audio attacks, the recorded audio will undergo an additional preprocessing phase before playback (e.g., computing the inverse MFCC [13], inverse filtering [74], or modulation onto ultrasonic carrier [85]). In contrast, genuine speech commands are directly inputted into the intelligent audio system via one-time over-the-air propagation. The redundant procedures of audio attacks will introduce additional noises to the audio signal in several aspects: first, the attack audio propagates through physical environments twice, resulting in more distortions due to the effects of room acoustics (e.g., environmental noise, attenuation, and reverberation); second, the hardware imperfection (e.g., non-flat frequency response and



(a) Magnitude spectrum of outdoor (b) Magnitude spectrum of indoor #1 (c) Magnitude spectrum of indoor #2 (d) Magnitude spectrum of in-vehicle

(e) Phase spectrum of outdoor (f) Phase spectrum of indoor #1 (g) Phase spectrum of indoor #2 (h) Phase spectrum of in-vehicle

Figure 4: Illustration of the magnitude and phase spectrum of genuine and replayed audio in different environments.



(a) Magnitude spectrum of outdoor (b) Magnitude spectrum of indoor #1 (c) Magnitude spectrum of indoor #2 (d) Magnitude spectrum of in-vehicle

(e) Phase spectrum of outdoor (f) Phase spectrum of indoor #1 (g) Phase spectrum of indoor #2 (h) Phase spectrum of in-vehicle

Figure 5: Comparison of the genuine and replayed audio recorded from different channels of a microphone array in various environments.

noise in electronics) of the recording and playback devices will also greatly impact the signal received by the intelligent audio system.

3.2 Potential Feature Analysis

Traditionally, power spectrum-based features are the most widely used features for audio signal analysis, and their effectiveness on replayed audio detection has been validated by many prior studies [12, 24, 66, 76]. However, power spectrum-based features alone might not be sufficient, as a recent work [74] has demonstrated that these features can be potentially manipulated by sophisticated attackers to evade the detection. In addition to the widely-considered magnitude-based features, recent studies on single-channel replay attack detection have revealed that the phase domain features also contain complementary channel information to the magnitude-based features that are potentially useful for replayed and synthesized audio detection [37, 43, 55, 72, 79]. However, utilizing multi-channel phase information for audio attack detection remains unexplored. To investigate the discriminability of the magnitude and phase information derived from multi-channel audio, we perform a feature analysis on the recently-published ReMASC dataset [22], which contains genuine and replayed speech samples recorded from multi-channel devices in four environments. The dataset and its recording environments are detailed in Section 7. Specifically, we divide all speech samples recorded by the ReSpeaker 4-Mic Linear Array into 4 groups according to the recording environment (i.e., outdoor, indoor #1, indoor #2, and in-vehicle). This results in a total of 192, 713, 275, and 673 genuine speech samples and 311, 2157, 846, and 959 replayed speech samples for each environment, respectively. Figure 4 plots the average power spectrum for all the genuine and the replayed audio samples and the continuous phase spectrum averaged across all channels. From the figure, we can clearly observe that both the magnitude and phase spectrum exhibit distinguishable patterns between genuine and replayed audio in all the environments. This confirms that both magnitude and phase information in the frequency domain can be used to learn the innate difference between the vocalization mechanism of humans and loudspeakers.

Table 1: Inter-channel L1 distance of magnitude and phase spectrum in different environments.

		Outdoor						Indoor #1						Indoor #2						In-vehicle					
Channel Pair		1-2	1-3	1-4	2-3	2-4	3-4	1-2	1-3	1-4	2-3	2-4	3-4	1-2	1-3	1-4	2-3	2-4	3-4	1-2	1-3	1-4	2-3	2-4	3-4
Genuine	Magnitude (dB)	1.00	1.07	0.94	0.80	1.56	**1.71**	0.83	0.89	0.68	0.57	**1.22**	1.13	0.96	1.11	0.98	0.92	1.16	**1.26**	0.76	1.01	0.71	0.72	0.90	**1.05**
	Phase (radians)	202.19	276.56	**348.61**	74.49	147.42	75.86	10.55	15.54	35.18	18.55	27.18	**45.57**	8.35	14.54	**34.09**	14.61	26.97	24.53	27.15	**108.61**	70.84	82.08	44.68	40.38
Replayed	Magnitude (dB)	1.39	1.92	**2.09**	1.28	1.84	1.61	0.79	0.86	0.96	0.82	**1.25**	1.09	1.17	1.31	1.39	1.14	**1.46**	1.40	0.78	0.81	0.81	0.57	1.09	**1.09**
	Phase (radians)	82.38	**117.48**	81.45	35.11	13.82	36.058	15.15	22.79	79.60	35.76	**94.19**	60.86	46.22	31.35	16.00	15.45	**34.65**	23.81	57.86	47.56	**78.76**	16.13	21.33	31.21

Table 2: Channel-wise L1 distance of magnitude and phase spectrum between genuine and replayed audio in different environments.

	Outdoor				Indoor #1				Indoor #2				In-vehicle			
Channel#	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Magnitude (dB)	3.60	3.48	**3.79**	3.45	2.59	2.84	**2.95**	2.85	2.00	2.38	2.44	**2.48**	1.06	**1.10**	1.02	0.98
Phase (radian)	1587.52	1707.57	1746.88	**1854.19**	2933.04	**2940.81**	2925.86	2819.91	**878.47**	824.35	837.70	829.94	1458.11	1427.10	**1519.01**	1447.60

3.3 Distinct Information Carried on Multiple Channels

Conventional approaches for multi-channel speech recognition often rely on beamforming techniques [10] to combine the received multi-channel signal into an enhanced single-channel signal to separate or extract speech signals from noisy environments. In particular, a beamformer acts like a spatial filter to enhance the signal from a specific direction of interest (i.e., the speech signal) and reduce the contamination caused by signals from other directions (e.g., ambient noises). However, different from speech separation or speech recognition that focuses only on the speech signal, the multi-channel audio signal picked up by the microphone array could contain distinct information (e.g., different surrounding noise patterns) that are beneficial to the machine-induced audio detection process. To validate the feasibility of leveraging multi-channel audio to enhance the performance of audio attack detection, we further analyze the ReMASC dataset by plotting the magnitude and phase spectrum of two individual channels, as shown in Figure 5. We observe that each individual channel possesses unique information (especially visible in the phase domain) that could be helpful for the detection of machine-induced audio. In addition, we perform statistical analysis on the audio samples to quantify the magnitude/phase difference between each pair of channels by calculating their average L1 distance. As shown in Table 1, there exists a difference in both magnitude and phase between any two channels of the recorded audio, showing that each channel indeed carries distinct information. Moreover, we observe that the pair of channels with the most distinct magnitude information does not necessarily carry the most distinct phase information. These findings encourage us to design a deep learning model that leverages both the magnitude and phase information of multi-channel audio and extracts features from each channel independently to achieve robust and high-performance audio attack detection.

3.4 Dominant Channel in Each Environment

To further investigate the impact of different acoustic environments, we compute the channel-wise L1 distance of magnitude and phase spectrum between genuine and replayed audio for each environment in Table 2, where the dominant channel that carries the most discriminative information for detecting replayed audio is marked in bold. We observe that the dominant magnitude/phase channel varies in different acoustic environments, which is caused by the varying recording condition (e.g., environmental noise) and behavior of the recording and playback device (e.g., the relative location of the sound source to the microphone array). The results demonstrate the characteristics of the genuine and replayed audio can be heavily affected by the type of recording environment. As a result, the patterns learned from existing environments might not generalize to new environments, which motivates us to explore a way to remove environment-specific features from the model.

4 SYSTEM DESIGN

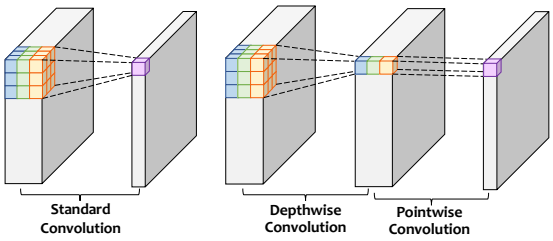
4.1 Design Objectives and Challenges

We aim to build a holistic solution to detect all the audio attacks induced by machines. Specifically, the solution needs to meet the following design objectives: 1) the model should be able to utilize the rich information encoded in multi-channel audio to achieve enhanced audio attack detection accuracy compared to existing single-channel based methods; 2) in order to build a holistic defense against any machine-induced audio attack, the model should be able to capture a set of generic acoustic features that distinguish genuine speech from machine-induced audio; 3) the model should rely on environment-independent features only to maintain a decent detection performance in different acoustic environments.

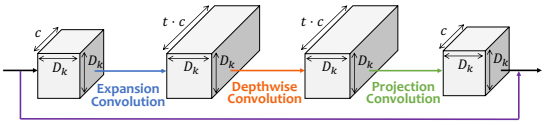
Challenges. To design such a holistic and robust system, we have to address the following challenges: 1) The voice interface embedded in intelligent audio devices requires a swift system response for usability considerations. To achieve timely detection, the audio attack detection system should be able to make a decision relying on only a short fraction of audio (e.g., ≤ 1 second); 2) The attack audio may be induced by disparate types of loudspeakers (e.g., standalone loudspeaker, built-in speaker on smartphone, and ultrasound speaker) that have varying frequency responses. Therefore, the model needs to be able to capture general features that are pervasive across all playback devices; 3) Explicitly collecting labeled data for all common acoustic environments is rather difficult in practice and thus it will be more desirable to enable the model to generalize to new acoustic environments without requiring labeled data for achieving robust defense.

4.2 Prepossessing

In real-world application scenarios, the detection model should be able to make a decision relying on a short segment of the streaming audio recorded by the microphone array. The length of the segment l should be set as short as possible to achieve timely detection for



(a) Standard convolution (b) Depthwise separable convolution



(c) Inverted residual block

Figure 6: Illustration of the standard and depthwise separable convolution and the inverted residual block.

real-time applications while maintaining a high recognition accuracy. In our implementation, we choose to set l to be 1 second. This gives us a multi-channel audio signal segment of size $l \times c$, where c is the number of channels for the microphone array. Instead of directly operating on the raw waveforms, we utilize the audio signal to create a unified time-frequency map from the power magnitude obtained via Short-time Fourier transform (STFT), which shows how the intensity of each frequency component changes over time. Specifically, we keep the entire audio signal segment without applying voice activity detection and use a sliding window to process the signal into overlapping frames with a frame length of 10 ms and a step of 5 ms. We then apply the Hann window function [40] to each frame and calculate its spectrum using STFT, with the length for fast Fourier transform being set to 512. For audio signals sampled at the rate of 44.1kHz, the resulting dimension of the time-frequency maps for each audio channel is 199 (time domain) by 257 (frequency domain). Similarly, we process the phase information into time-frequency maps of the same size by computing the angle of the complex STFT values in radians and stack the phase maps along with the magnitude maps. The final feature map shape used as the input for our network is $199 \times 257 \times 2c$ for a c-channel audio signal.

4.3 Multi-channel Replay Attack Detection Network

After preprocessing, the audio signals will be processed into image-like feature maps, which enables us to leverage the rich body of research on convolutional neural networks (CNNs) in the computer vision domain to guide the design of our audio attack detection network. Specifically, we propose to explore two flavors of network configuration according to different usage scenarios: (1) *Type I*: a large and powerful network that has more representational capacity to enable high attack detection accuracy for desktop or cloud application; and (2) *Type II*: a fast and lightweight network that provides more computational and energy savings, which makes it suitable for mobile and IoT deployments.

Design of *Type I* Network. Inspired by the previous study [50] on CNN architecture for processing image data, we configure our *Type I* network using modules composed of stacked convolution layers with small-sized filters (e.g., 3×3) and pooling layers. The intuition is that compared with large convolution filters, stacked small convolution filters can achieve the same effective receptive field as larger layers but with a fewer number of parameters (e.g., two stacked 3×3 layers have an effective field of 5×5, while three stacked 3×3 layers have an effective field of 7×7), which makes the model smaller and easier to be optimized. Additionally, decomposing one convolution layer with a large filter into multiple layers unlocks additional layers of non-linearity by injecting more non-linear activation functions (e.g., Rectified Linear Unit (ReLU)), which helps the network to capture complex patterns in the data.

Design of *Type II* Network. For our *Type II* network design, we adopt the architectures proposed in MobileNet [28, 44] to compress the model size and achieve efficient detection while maintaining relatively high detection accuracy. The key innovation of MobileNet compared with traditional deep networks (e.g., GoogLeNet [51], DenseNet [29] and ResNet [26]) is the usage of depthwise separable convolution and the bottleneck residual block, which aims to replace the expensive standard convolution layers with depthwise separable convolutions which require a much fewer number of parameters. As shown in Figure 6(a) and 6(b), the standard convolution operation is substituted with a combination of two different convolution operations, i.e., a depthwise convolution and a pointwise operation. Different from standard convolution that combines all input channels, depthwise convolution performs convolution on each channel separately. The output channels of the depthwise convolution operation are then combined using a pointwise convolution with 1×1 kernels. For a convolution operation with M input channel, N output channel, and $D_k \times D_k$ kernel, this transformation significantly reduces the computational cost by a factor of $\frac{1}{N} + \frac{1}{D_k^2}$, which is especially helpful for processing multi-channel audio signals that have a large number of input channels (e.g., a 6 channel audio signal will produce a 12 channel input feature map). Leveraging this depthwise convolution, we can further construct inverted residual blocks (Figure 6(c)) by adding expansion layers that expand the compressed low-dimensional representation to high-dimensional space and projection layers that project the filtered representation back to low-dimensional subspace. The expansion ratio t is used to control how much the representation is expanded. In addition, a residual shortcut connect is added between the blocks to help accelerate the optimization process. A width multiplier hyperparameter is used to further scale the model by increasing/reducing the number of channels for all layers by a factor of α.

Network Structure. The overall structure of the proposed audio attack detection network is presented in Figure 7. Specifically, the network is composed of 3 components: a CNN feature extractor, a fully-connected (FC) genuine/attack audio classifier, and an optional domain discriminator which is only involved in environment-independent training and will be detailed in Section 5. The *Type I* network is built upon the VGG-16 network [50], with the number of input channels modified according to the multi-channel audio and the number of output neurons set to 2. We build the *Type II* network based on the MobileNetV2 [44], with similar modifications made to the network structure to accommodate the multi-channel audio attack detection task.

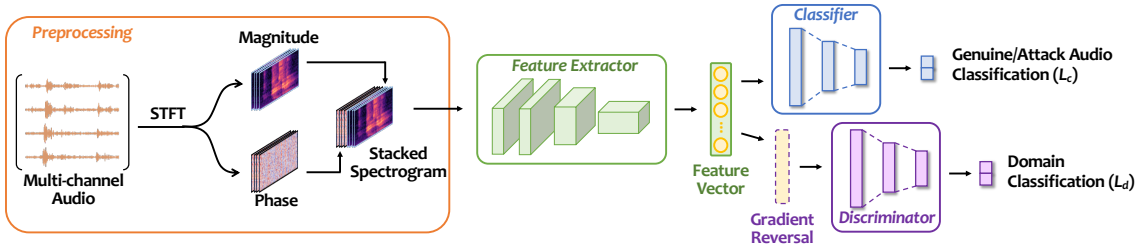


Figure 7: Illustration of the proposed audio attack detection network.

4.4 Optimization

The attack detection is modeled as a binary classification problem (i.e., genuine speech or machine-induced audio) and the networks are trained in an end-to-end manner from raw waveforms of multi-channel audio to the prediction label, with the preprocessing units (magnitude and phase spectrogram extraction) being implemented as part of the network. We use cross entropy as the classification loss to train the network. Due to the difficulty in gathering large sets of human voice samples, public audio attack datasets often suffer the class imbalance problem where the data distribution is biased towards the attack audio class (e.g., the ratio of genuine audio to replayed audio in the ReMASC dataset [22] is approximately $1:5$). This poses a challenge for the deep learning model training as the minority class (i.e., the genuine speech) is more important and thus more sensitive to classification errors. To address the class imbalance problem, during training we re-weight the cross-entropy for each class according to the number of samples available in the training set. The ADAM optimizer [31] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ is used to train the network for a total number of 100 epochs. As for learning rate schedule, the step learning rate decay with warm-up is used, where the learning rate is initially set to a small value and increased by $10\times$ in the first 20 epochs and then reduced by a half every 20 epochs. Batch normalization layers and l2 weight regularization are also applied to stabilize the training process and prevent over-fitting.

5 DOMAIN-INVARIANT REPRESENTATION LEARNING

The learning-based predictive modeling approaches heavily rely on the training data to make predictions and its performance is likely to degrade if the provided training samples are not an accurate reflection of the underlying distribution of actual data. This poses a challenge for audio attack detection since the model will inevitably face new acoustic environments that are unrepresented in the training data when deployed in practice.

To address this problem, we take inspiration from the recent success in domain adaptation techniques in the computer vision domain [20, 47, 67] and adopt an unsupervised domain adaptation scheme for achieving domain-invariant representation learning. In the context of audio attack detection, the term "domain" refers to different acoustic environments and the domain adaptation process aims to help the model generalize from the training-time environment (i.e., the source domain) to the test-time environment (i.e., the target domain). Specifically, let w_f and w_c denote the parameters of the feature extractor and the classifier, respectively. As is mentioned in Section 4, the network is trained on the classification loss $L_c(w_f, w_c)$ to recognize genuine/attack audio. To help the feature extractor learn domain-invariant features, we introduce a new domain discriminator with parameters w_d during training. The domain discriminator shares the same architecture as the classifier but the objective of the discriminator is to distinguish between the source domain training samples and the target domain training samples by minimizing the domain classification loss $L_d(w_f, w_d)$. The objective of the domain-invariant training process is to search for the parameter set w_f to minimize the audio classification loss L_c and simultaneously maximize the domain classification loss L_d, which can be achieved by minimizing the following integrated loss function:

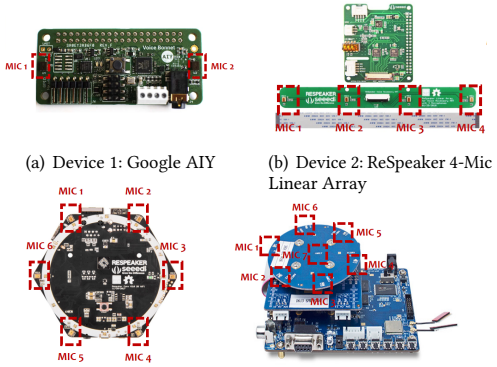
$$L(w_f, w_c, w_d) = L_c(w_f, w_c) - \lambda \cdot L_d(w_f, w_d),$$

where λ is a weighting factor to control the impact of the domain discriminator on the learned feature mapping during training. This can be achieved by inserting a gradient reversal layer [20] into the network which implements the identity function during forward propagation and reverses the gradient by multiplying it by a negative scalar (i.e., λ) during the backpropagation process. After training, the feature extractor will learn to extract features that are both discriminative for detecting various audio attacks and invariant to the change of acoustic environments.

6 ATTACK IMPLEMENTATION

In order to evaluate our designed machine-induced audio attack detection approach, we reproduced a set of representative audio attacks. In addition to conventional replay attack, for which we used a recently-published dataset (Section 7), we generated a set of adversarial speech samples through the following procedures and conducted extensive real-world experiments in various environmental conditions.

Modulated Replayed Attack. Due to the security concerns brought by replay attacks, a number of defense approaches have been developed to detect replayed audio signals, by examining unique acoustic distortions (e.g., energy distribution in the frequency domain) induced by the playback device. To bypass such defenses, a recent study [74] designed a new type of replayed attack, namely modulated replayed attack, which can compensate for the acoustic distortions through profiling the frequency response of the playback device. Specifically, in our implementation, the frequency response is measured with 68 single-frequency testing signals across $0 \sim 4000Hz$. We play the testing signals on three playback devices (i.e., Huawei Nova 4, iPhone 12 Pro Max, and HP Elitebook 1050 G1 laptop) and record the replayed audio signals with a microphone (i.e., ReSpeaker Core v2.0). We then use the played testing signals and the recording to generate an inverse



(a) Device 1: Google AIY (b) Device 2: ReSpeaker 4-Mic Linear Array

(c) Device 3: ReSpeaker Core V2 (d) Device 4: Amlogic A113X1

Figure 8: Microphone arrays used for data collection.

filter [74] to compensate for the acoustic distortions for each playback device. Finally, we record the 10 original voice commands shown in Table 9(a) spoken by a volunteer and pass the recordings through the corresponding inverse filter to generate modulated speech samples for each playback device.

Synthesis Attack. Synthesis attacks usually rely on a speech synthesis model to produce attacking audio mimicking the voice characteristics of the victim. The current synthesis models based on deep learning can simulate natural sounding voices similar to human subjects. To evaluate synthesized speech, we use two state-of-the-art speech synthesis models, including Google Text-to-Speech based on WaveNet [1, 69] and Tacotron 2 based on WaveGlow [46]. Both WaveNet and WaveGlow are CNN-based audio generative models exploiting temporal dependencies for speech signal generation. For the Google Text-to-Speech, we directly use a pre-trained WaveGlow model of a male speaker provided by the API, while for Tacotron 2, we train a WaveGlow-based speech synthesis model by using 13,100 voice samples from a female speaker (i.e., LJ Speech Dataset [3]). We use the two models to separately generate the 10 original voice commands listed in Table 9(a).

Hidden Voice Command. Hidden voice commands [13, 68] are obfuscated voice commands that are unintelligible to human beings but can be interpreted by intelligent audio systems. Such attacks exploit the perception difference between humans and machines (e.g., speech recognition models) in processing speech and modulate the recorded voice samples into attacking audio. To generate hidden voice commands, the attack will first extract voice features from normal commands and then train a network for reconstructing voice with these features and meanwhile continuously update parameters of the network and feature extraction to make it unintelligible to humans. The attack can be either black-box (through inverting MFCC features) or white-box (through applying gradient descent-based approach on a target speech recognition model). A recent study even proposed a more practical hidden voice commands [7] aiming to spoof the feature extraction process of speech recognition models, rendering the attack black-box and effective. To evaluate our system, we use 14 publicly released hidden voice commands, including 10 regular hidden voice commands [2] and 4 practical hidden voice commands [4].

Table 3: Description of the collected audio attack datasets.

Type of Audio	Environment	Distance (cm)	# Samples of Device 1	# Samples of Device 2	# Samples of Device 3	# Total Samples
HVC	Room 1, 2, 3	30, 50, 100, 200, 300	1,812	1,680	1,478	4,970
Synthesis	Room 1, 2, 3	30, 50, 100, 200, 300	2,397	2,531	2,531	7,459
Inaudible	Room 1, 2, 3	10, 30	520	520	520	1,560
Adversarial	Room 1, 2, 3	30, 50, 100	503	503	503	1,509
Modulated	Room 1, 2	30, 50, 100	510	510	510	1,530
Genuine	Room 1, 2	50, 100, 200, 300	1,324	1,147	1,240	3,711

Inaudible Attack. The adversary can launch inaudible attacks by modulating the voice commands into ultrasound frequency bands [41, 85] (e.g., over $20kHz$). Although ultrasound signals cannot be perceived by the human ear, they can be demodulated by the microphones in audio intelligent devices due to their inherent non-linearity. To implement inaudible attacks, we first use Google Text-to-Speech API to generate the 10 original voice commands listed in Table 9(a). We then use amplitude modulation to modulate the voice commands onto a baseband signal of $35kHz$, where the modulated sound is completely inaudible and can be demodulated by microphones. The modulated signals are generated on a Keysight 33500B signal generator and played by an ultrasonic speaker (Avisoft Bioacoustics Vifa [5]).

Audio Adversarial Example. The current intelligible audio systems mainly rely on deep neural networks to perform speech recognition, which are inherently vulnerable to well-craft and imperceptible adversarial perturbations [14, 35, 84]. The adversary can inject the adversarial perturbations into the audio signals to spoof the deep learning models. We implement the gradient-based perturbation generation presented in the previous study on audio adversarial examples [14], which targets to spoof an end-to-end speech recognition (i.e., DeepSpeech [25]). To implement the attack, we first generate 10 original voice commands (i.e., original commands listed in Appendix Table 9(a)) using Google text-to-speech API and then compute the adversarial perturbations to fool DeepSpeech with the corresponding target commands (i.e., target commands listed in Appendix Table 9(b)). The perturbations are then added to the original voice commands for the attack.

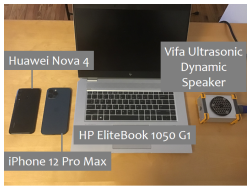
7 PERFORMANCE EVALUATION

7.1 Experimental Methodology

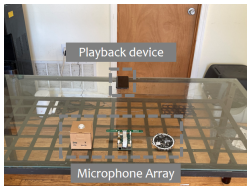
To evaluate our system under the replay attack, we use a public dataset collected using 4 different microphone arrays. For more advanced audio attacks (e.g., hidden voice commands, inaudible attacks), we use 3 representative microphones arrays shared in the public dataset to recorded the attacking sound and genuine human speech for evaluation.

7.1.1 Public Replay Attack Dataset. To evaluate our system under the replay attack, we use a public dataset, ReMASC [22], which is collected using a set of 4 microphone arrays with $2 \sim 7$ audio channels. We partition the dataset into the core training set and the evaluation set as described in ReMASC. The training set and the evaluation set are disjoint and contain 26, 946 and 17, 581 audio samples, respectively.

- **Devices.** To mimic multi-channel recording in commercial intelligent audio devices, ReMASC uses 4 microphone arrays with different number of audio channels as shown in Figure 8. These microphone arrays include: 1) Google AIY Voice Kit (2 channels); 2) ReSpeaker 4-mic linear array (4 channels); 3) ReSpeaker Core V2 (6 channels) 4) Amlogic A113X1 (7 channels). To generate

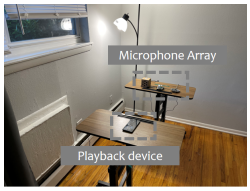


(a) Playback devices



(b) Room 1

(c) Room 2



(d) Room 3

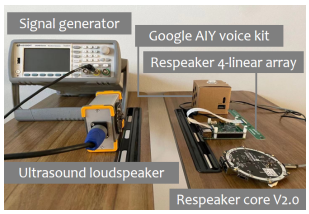
Figure 9: Recording devices and environments of the self-collected dataset.

the attacking sound, ReMASC uses 3 different playback devices, including a Sony SRSX5, a Sony SRSX11, an Audio Technica ATH-AD700X headphone, and an iPod Touch.

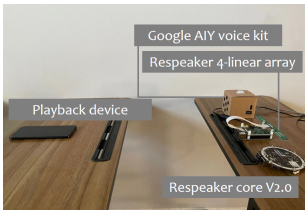
- **Environments.** The dataset contains a total number of 9,240 genuine speech samples and 45,472 replayed recordings collected in 4 different acoustic environments: 1) Outdoor (Env-A): an outdoor student plaza with various background noises such as chatting, traffic, and wind; 2) Indoor #1 (Env-B): a quiet study room; 3) Indoor #2 (Env-C): a lounge with music players and TVs ruining in the background; and 4) In-vehicle (Env-D): inside a moving vehicle (Dodge Grand Caravan) in different areas (e.g., campus, residential area, urban area, and highway) with speeds ranging from 3 to 40 miles per hour. The samples are recorded at varying distances (0.5 − 6m) and angles (0 − 90 degrees) according to each environment. The data volume and the involved number of speakers for each environment are shown in Appendix Table 10.

7.1.2 Self-collected Audio Attack Dataset. Besides conventional replay attacks, we also collect data samples of other 5 more advanced audio attacks using multiple microphone arrays in different environments, following the implementation described in Section 6. Table 3 shows the detail of the self-collected dataset.

- **Devices.** The data are collected using three microphone arrays, i.e., Google AIY voice kit, ReSpeaker 4-Mic linear array, and ReSpeaker core V2, which are shown in Figure 8. For the inaudible attack, we use an ultrasound speaker (i.e., Vifa Ultrasonic Dynamic Speaker), while for other audio attacks, we use 2 smartphones (i.e., Huawei Nova 4 and iPhone 12 pro max) and a laptop (i.e., HP EliteBook 1050G1) as the playback device, as shown in Figure 9(a).
- **Environments.** The attack audio and genuine speech samples are collected in 3 different room environments as shown in Figure 9(b)-(d), including two living rooms and a bedroom.
- **Genuine Speech and Attack Setup.** We recruit 6 participants (i.e., 4 males and 2 females) aging from 22 and 30 to collect the genuine speech samples. The attacking audio and genuine speech are mostly recorded at 3 different distances between the participant/loudspeaker and the microphone arrays, i.e., 30*cm*, 50*cm*, 100*cm*, except for the inaudible attack, which is only recorded at 10*cm* and 30*cm* due to its short effective range [85]. Moreover, for genuine speech and attacks that are less sensitive to the recording distance (i.e., the hidden voice command and synthesis attack), we further conduct experiments to collect samples in the long-range scenario by extending the attack range to 200*cm* and 300*cm*. The device placement for recording the attack audio is shown in Figure 10.



(a) Setup for inaudible attack



(b) Setup for the remaining audio attacks

Figure 10: Experimental setup of audio attacks.

Table 4: Results for replay attack detection in environment-dependent settings.

EER(%)/RA(%)	Device 1	Device 2	Device 3	Device 4
Gong *et al.* [23]	14.9/-	15.4/-	16.5/-	19.8/-
CQT-LCNN [54]	15.0/90.3	23.4/76.3	26.9/77.9	21.5/81.0
LFCC-LCNN [62]	14.8/89.8	24.2/90.0	23.5/81.4	27.3/78.6
RawNet2 [59]	10.8/89.5	15.9/89.3	18.2/82.9	24.9/74.2
Ours (*Type I*)	**6.6**/**96.0**	**11.0**/**92.8**	**9.2**/**93.5**	**15.7**/85.4
Ours (*Type II*)	10.3/94.2	14.6/91.8	12.3/92.6	18.2/**90.3**

7.1.3 Evaluation Metrics. We use two metrics to evaluate the performance of the system: (1) *Recognition Accuracy (RA)*: Audio attack detection can be viewed as a binary classification problem. Recognition accuracy is the percentage of audio samples being correctly classified; and (2) *Equal Error Rate (EER)*: EER is a commonly used metric for evaluating replay attack detection system [32]. It depends on two detection error rates: the false acceptance rate (FAR) and the false rejection rate (FRR). EER corresponds to the point at which the two detection error rates are approximately equal.

7.1.4 Baseline Models. We compare our results with 4 state-of-the-art baseline replay attack detection models: (1) *Gong et al.* [23], which is a multi-channel replay attack network composed of learnable filter-and-sum beamformer, a frequency convolution layer, and multiple stacked LSTM layers for classification; (2) *CQT-LCNN [54]*, which is a single-channel replay attack detection model using the log power magnitude spectrum extracted via the constant Q transform (CQT) [66] as the features and a light convolutional network (LCNN) [63] as the classifier. This single model achieves 1.23% EER in the ASVspoof2019 Physical Access (PA) [60] scenario and can be further improved to 0.54% EER (ranks the 2*nd* place) if score-level fusion is applied using models with other front-end features; (3) *LFCC-LCNN [62]*, which is a single-channel-based model that adopts linear frequency cepstral coefficients (LFCC) as the front end and LCNN as the back-end classifier. This model is used as the official baseline for the ASVspoof2021 challenge [6]; and (4) *RawNet2 [59]*, which is a single-channel model that aims to release the constraints of hand-crafted features by training an end-to-end CNN-GRU network with sinc-convolution layer [56] to extract useful cues directly from raw audio waveforms. SVM-based fusion of RawNet2 and high-spectral-resolution LFCC [58] can achieve

Table 5: Results for replay attack detection in the environment-independent settings[3].

EER(%)	Device 1			Device 2			Device 3			Device 4		
Models	Gong *et al.* [23]	Ours (w/o DA)	Ours (w/ DA)	Gong *et al.* [23]	Ours (w/o DA)	Ours (w/ DA)	Gong *et al.* [23]	Ours (w/o DA)	Ours (w/ DA)	Gong *et al.* [23]	Ours (w/o DA)	Ours (w/ DA)
Env-A	35.2	31.2	**22.8**	34.6	16.2	**12.5**	23.8	21.1	**14.5**	31.5	24.5	**19.1**
Env-B	-	-	-	36.4	17.7	**11.9**	40.4	36.4	**27.4**	44.9	42.6	**39.0**
Env-C	36.7	38.3	**27.0**	18.5	24.0	**8.8**	13.0	14.2	**12.6**	32.7	28.6	**19.9**
Env-D	34.0	37.4	**27.4**	41.0	38.9	**26.3**	43.6	41.2	**23.8**	40.6	41.7	**34.1**

an EER as low as 1.12%, which ranks the *2nd* best place in the ASVspoof2019 Logical Access scenario (LA).

7.2 Overall System Performance for Replay Attack

We first evaluate the overall performance of the proposed system on replay attacks using the public ReMASC dataset. For fair comparison, we use the same default data separation scheme suggested in the original paper [22] for all baseline methods and develop a separate model for each recording device as is used by Gong *et al.* [23]. For the *Type II* network, we use a width multiplier of $\alpha = 1$ for device 1 & 2 and $\alpha = 1.5$ for device 3 & 4. Each model is trained using a batch size of 32 for 100 epochs on the same learning rate scheduling strategy with an initial learning rate of 1×10^{-3} for our *Type II* models and 1×10^{-5} for other models. We implement all baseline methods and compare the experiment results with the proposed models in Table 4. By default we use the signals collected from the first channel for the training of single-channel-based models. For the beamforming-based network proposed by Gong *et al.* [23], we report its best results presented in the original paper (RA is not shown since it has not been reported). From the results we can see that RawNet2 achieves the overall best performance among all single-channel-based methods, which is even able to produce EER that is lower than the multi-channel beamforming-based network proposed by Gong *et al.* [23] for recording device 1 with 2 channels. However, in general we observe that multi-channel-based methods still outperform single-channel-based methods with the performance gain becoming more visible as the number of available channels increases. The proposed *Type I* network can consistently achieve better EERs that are 20% − 55% lower comparing to the existing beamforming-based network. The *Type II* network can also reduce the EER by up to 31% compared to the beamforming-based network. These results verify that comparing to using a beamformer to combine multi-channel audio signal into an enhanced signal, utilizing magnitude and phase information from all available channels can result in better performance in audio attack detection.

Inference Time. The inference time is crucial for real-time detection. To investigate the inference time of our models, we run experiments on a Nvidia 2080Ti GPU with a batch size of 16 and repeat for 100 trials to measure the average inference time. The results show that the proposed *Type I* model takes 36.5ms while the *Type II* model only takes 23.3ms. Compared to the latency of commercial speaker recognition model [52] (~40ms) and speech recognition model [61] (~600ms), the latency of the proposed detection model is sufficient to achieve timely detection of any types of audio attacks for various real-time applications.

[3]The dataset lacks genuine speech samples recorded using Device 1 in Env-B, and therefore the EER cannot be obtained.

Table 6: Comparison with single-channel-based methods in the environment-independent settings on Device 2.

EER(%)	Gong *et al.* [23]	CQT-LCNN [54]	LFCC-LCNN [62]	RawNet2 [59]	Ours (w/o DA)	Ours (w/ DA)
Env-A	34.6	30.8	40.3	39.1	16.2	**12.5**
Env-B	36.4	43.4	37.8	26.4	17.7	**11.9**
Env-C	18.5	40.0	35.3	36.3	24.0	**8.8**
Env-D	41.0	31.9	57.4	39.1	38.9	**26.3**

Model Size. For desktop or cloud applications with sufficient storage and computational resources, we often prioritize the performance over the size of the model. However, for mobile and embedded applications that require the model to be executed offline in an on-device manner, the size of the model should be small enough in order to match the resource restrictions (e.g., memory, computational resource, and power consumption). The model size of our *Type I* network is around 479MB, which we believe can be deployed in most desktop or cloud applications. For our *Type II* network, the model size is only 18MB with $\alpha = 1$ and 40MB with $\alpha = 1.5$. This demonstrates that by utilizing the inverted residual module, the proposed *Type II* network is extremely lightweight while still retaining sufficient representation power to achieve a high attack detection accuracy.

7.3 Environment-independent Detection

In addition to inspecting the overall performance of the model by training on a mixture of data samples from all the environments, we also evaluate the model in environment-independent conditions. Specifically, we set one of the 4 environments to be the target domain for testing, while the remaining 3 environments serve as the source domain for training. We set $\lambda = 0.33$ and use the optimization techniques as mentioned in Section 4.4 to train the models. To validate the effectiveness of the domain adaptation (DA) training procedure, we compare the performance of our Type-I model with DA to the performance of model without DA in Table 5, where the results of the multi-channel beamforming-based network [23] are also shown for comparison. In addition, we compare the results of our models with single-channel-based methods using the data recorded from Device 2 in Table 6.

From the results we observe that models trained on data from source environments generally suffer low generalizability to new environments. In particular, Env-D (i.e., the in-vehicle environment) is the most difficult environment for the model to generalize among environment-independent cases. This is because the in-vehicle setting has several unique acoustic features (e.g., loud road and engine noises and strong reverberations due to narrow cabin) that cannot be learned from other environments. As shown in Table 6, apart from RawNet2 which achieves an EER of 26.4% in Env-B, all single-channel-based methods perform poorly in the environment-independent scenario (>30% EER). Despite this, leveraging the domain adaptation process, the proposed network is still

Table 7: Results for detecting replay attack as well as 5 other types of advanced machine-induced audio attacks.

EER(%)/RA(%)	Device 1	Device 2	Device 3
Gong *et al.* [23]	19.3/84.1	19.8/87.5	15.1/83.8
CQT-LCNN [54]	18.7/87.6	20.7/84.3	23.1/81.9
LFCC-LCNN [62]	19.5/85.2	24.2/81.2	26.7/81.3
RawNet2 [59]	19.0/87.4	23.9/83.9	23.5/84.0
Ours (*Type I*)	**13.1/91.5**	**15.2/90.6**	**10.3/92.3**
Ours (*Type II*)	15.4/88.3	15.5/90.3	14.9/90.2

able to reduce the EER by up to 42.2%, achieving an average EER of 21.8%, which is much lower comparing to the beamforming-based network (33.8%) and the proposed network (30.3%) without applying domain adaptation.

7.4 Robustness Against Other More Advanced Attacks

In this section, we expand the evaluation of the system robustness to include other more advanced audio attacks using the self-collected dataset described in Section 7.1.2. We randomly split the collected audio attack samples into training and test sets with 60% samples used for training and 40% samples reserved for testing, which is similar to the train/test split ratio used in the ReMASC dataset[4]. The separated datasets are then merged with the the audio samples from the ReMASC dataset, resulting in a total number of 9596, 10260, 9931 samples for training and 5816, 7295, 6951 for testing, for the device 1, 2, 3, respectively.

Table 7 compares the results of the proposed model with baseline models. As we can see, when considering all 6 types of audio attacks, the performance of most models are degraded compared to the ones that are trained exclusively for detecting replay attacks, due to the highly varying behaviors of the advanced audio attacks. The proposed *Type I* network, however, is still able to achieve the best performance among baseline models, achieving an overall average EER of 12.9% across all 3 devices. Despite its compact model size, the proposed *Type II* network is also able to achieve a relatively high recognition accuracy and low EER that surpasses existing single- and multi-channel-based models in most cases. These results demonstrate that the proposed methods can learn general features that are able to distinguish machine-induced audio from genuine speech to achieve robust detection of various types of audio attacks.

7.5 Ablation Study

Impact of Involved Channels. To investigate the impact of the number of involved channels on the detection performance, we train a group of models on the genuine and replayed audio data recorded using device 2 by varying the number of input audio channels and measure the resulting recognition accuracy and EER. From the results shown in Figure 11, we observe that the recognition accuracy increases as more channels are involved. The EER has decreased from 17.1% to 11.0% if using four channels. These results validate the effectiveness and benefits of using multiple channel audio for audio attack detection.

Impact of Phase Information. To investigate the impact of the phase information on the system, we modify the structure of the proposed *Type I* network to only involve magnitude spectrograms and evaluate its impact on the models' performance. We get a recognition accuracy of 96.3%, 87.9%, 73.1% and EER of 8.6%, 11.4%, 29.2% for device 1, 2, and 3, respectively. Compared to the model that uses both magnitude and phase spectrogram as input, the average EER is increased by 38.8%. This result validates that the phase spectrogram can serve as complementary information in addition to the magnitude spectrogram to help further improve the performance of audio attack detection.

[4]The suggested data separation for the ReMASC dataset is shown in Appendix Table 11.

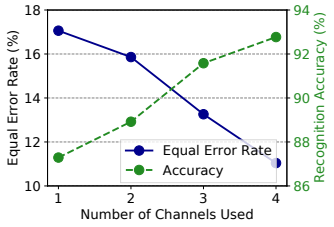


Figure 11: Results with different number of channels.

7.6 Model Interpretability Analysis

We further investigate the interpretability of our approach by visualizing the saliency map of the model and its learned representations.

Visualization of Saliency Map. We use the gradient-weighted class activation map (Grad-CAM) to visualize the decision-making process of the proposed deep learning model. Specifically, Grad-CAM uses the gradient of the target class to produce a localization map to highlight the important regions in the input feature map used by the model to make the prediction, allowing us to visualize the attention of the model. Figure 12 shows two examples of Gram-CAM generated from our model, where the three columns from left to right are the magnitude spectrogram of input, the generated CAM image, and the CAM overlaid on the spectrogram. We can observe that the most discriminative region on which the model mainly focuses is the low-frequency region, with some attention also being paid to the high-frequency noises. These findings are well-correlated with previous studies on discriminative frequency regions for replay attack detection [9, 12, 76], which demonstrates the effectiveness of the proposed learning-based approach.

Visualization of Learned Representations. To investigate the learned representations, we randomly select 20 audio samples from the genuine speech and each type of the audio attacks recorded using device 1 and compute the output of the first layer in the classifier of our *Type I* model as the embeddings. We first use Principle Component Analysis (PCA) [77] to reduce the dimensionality of each embedding to 100 and then use t-distributed Stochastic Neighbor Embedding (t-SNE) [70] to visualize the embeddings on a 2D plane. The visualization result is shown in Figure 13. From the figure, we can see that the genuine and attack audio samples are well-clustered, which verifies the model's ability to extract discriminative features. In addition, although the model is not trained to distinguish different audio attacks, we are still able to observe some patterns between different types of attacks. In particular, among all considered attacks, synthesis attack can produce audio samples that are the closest to genuine speech in the learned manifold, which shows that the deep learning-powered speech synthesizer used in our attack implementation can generate lifelike speech samples that resemble human speech.

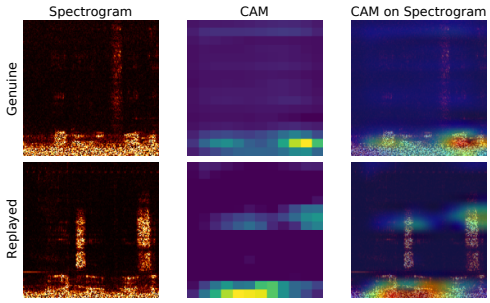


Figure 12: Visualization of the discriminative regions using Grad-CAM [45].

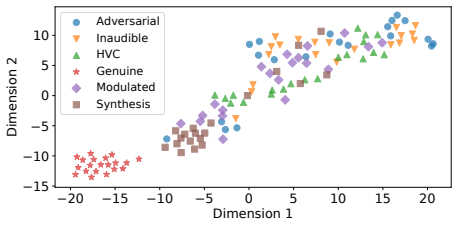


Figure 13: Visualization of the learned representation using t-SNE [70].

8 DISCUSSION

Integration with Intelligent Audio Systems. The developed audio attack detection models can be integrated into commercial intelligent audio systems with few or no modifications required on the hardware. Besides standalone intelligent audio systems (e.g., smart speakers), mobile intelligent audio systems such as smartphones also come equipped with multiple microphones for stereo recording and noise/reverberation cancellation. Since the microphone array on smartphones (usually located at the top and bottom of the phone frame) is of similar dimension with the 2-channel device (Google AIY voice kit) used in our experiments, we expect that our model can be easily adapted to smartphone use. Moreover, the model can be simply inserted at the beginning of the inference pipeline to check the legitimacy of the audio input before it reaches the speech or speaker recognition model. Depending on the application scenario as well as the capability of the system, the detection process can be executed either via cloud-based services or directly on the device. The proposed *Type I* model is desirable for cloud servers with sufficient computational power for achieving the maximum detection accuracy. In addition, the model can be used in parallel with other optimization components (e.g., attention module) and neural-based countermeasure models to potentially improve performance. For systems that have limited communication bandwidth or scenarios with rigorous privacy requirements, the data can also be processed locally. To support on-device inference on voice-controllable mobile and IoT devices that have constrained storage and computational resources, the audio attack detection model should be as compact and energy-efficient as possible. In this study, we provide a fast and lightweight *Type II* network, which is approximately 12× lighter and 1.5× faster compared to the proposed *Type I* network with slightly compromised performance as an option for resource-constrained devices. For future work, model compression [18, 38] and acceleration [16, 34] techniques can be adopted to further improve the efficiency of the model.

Potential Evasion. Grounded on a data-driven approach, the effectiveness of the proposed model requires collecting samples from existing audio attack methods for attack profiling. Thus, a sophisticated attacker with the ability to access the established profile may leverage this knowledge to design adaptive attacks to bypass the system. For instance, crafting an attack audio sample that is close to the genuine audio samples in the learned representation space may force the model to falsely accept it. However, launching such an adaptive attack in practice still faces several challenges. First, the attacker cannot directly regulate the received signal in its digital form as the model only accepts signals received through the physical channel (i.e., picked up by the microphone array) as valid inputs, while propagating in the physical environments will inevitably leave a certain level of traceable patterns to the audio signal. Although injecting signals via other modalities such as laser [57] may alleviate the distortion incurred by the over-the-air propagation, such attack only injects signal to one microphone channel at a time and therefore can be defended by cross-checking signals from all microphone channels before executing the command. An attacker may also attempt to evade detection by manipulating the sound field with multiple playback devices (e.g., a pair of stereo loudspeakers or a multi-channel surround sound system) to control the received signal in each microphone [64, 65]. However, precise manipulation of the signal received by each microphone channel is hard to achieve due to the low directionality and diffraction of sound. In addition, such attacks still involve the same recording and playback process which will cause distortions to be projected into the magnitude and phase domain. Moreover, solely defeating the detection model isn't sufficient. Since our model is proposed as an add-on module before the actual audio processing model (e.g., speech or speaker recognition model), the attacker needs to bypass both models to achieve a successful attack, which remains challenging in practice.

9 CONCLUSION

In this paper, we propose a holistic solution for detecting machine-induced audio attacks by leveraging the readily available microphone array on modern intelligent audio systems. We utilize the magnitude and phase information derived from multi-channel audio and train a deep learning model to capture the fundamental difference between human speech and adversarial audio launched from playback devices. To improve the generalizability to new acoustic environments, we use unsupervised domain adaptation to help the model learn to extract domain-invariant features. We also develop a more compact model that's suitable for resource-constrained mobile and IoT devices. Extensive experiments on a public multi-channel replay attack dataset and a self-collected advanced audio attack dataset show that the proposed method can achieve an EER as low as 6.6% for detecting a variety of audio attacks and still maintains a relatively high recognition accuracy even in the challenging environment-independent case.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their insightful feedback. This work is supported in part by National Science Foundation grants CNS2114161, CNS2114220, CCF1909963, and CCF2028876, and Air Force Research Lab grant FA87501820058.

REFERENCES

[1] 2020. Google Text-to-Speech. (2020). https://cloud.google.com/text-to-speech/docs

[2] 2021. Hidden Voice Commands. (2021). https://www.hiddenvoicecommands.com/demo

[3] 2021. The LJ Speech Dataset. (2021). https://keithito.com/LJ-Speech-Dataset/

[4] 2021. Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems. (2021). https://sites.google.com/view/practicalhiddenvoice/home

[5] 2021. Ultrasonic Dynamic Speaker Vifa. (2021). http://www.avisoft.com/playback/vifa/

[6] 2021. ASVspoof 2021: Automatic Speaker Verification Spoofing and Countermeasures Challenge Evaluation Plan. (2021). https://www.asvspoof.org/asvspoof2021/asvspoof2021_evaluation_plan.pdf

[7] Hadi Abdullah, Washington Garcia, Christian Peeters, Patrick Traynor, Kevin R. B. Butler, and Joseph Wilson. 2019. Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems. (2019). arXiv:cs.CR/1904.05734

[8] Anup Agarwal, Mohit Jain, Pratyush Kumar, and Shwetak Patel. 2018. Opportunistic sensing with MIC arrays on smart speakers for distal interaction and exercise tracking. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6403–6407.

[9] Muhammad Ejaz Ahmed, Il-Youp Kwak, Jun Ho Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. 2020. Void: A fast and light voice liveness detection system. In *29th USENIX Security Symposium (USENIX Security 20)*. USENIX Association, 2685–2702. https://www.usenix.org/conference/usenixsecurity20/presentation/ahmed-muhammad

[10] Jacob Benesty, Jingdong Chen, and Yiteng Huang. 2008. *Microphone array signal processing*. Vol. 1. Springer Science & Business Media.

[11] Logan Blue, Hadi Abdullah, Luis Vargas, and Patrick Traynor. 2018. 2ma: Verifying voice commands via two microphone authentication. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security (ACM ASIA CCS)*. 89–100.

[12] Logan Blue, Luis Vargas, and Patrick Traynor. 2018. Hello, is it me you're looking for? differentiating between human and electronic speakers for voice interface security. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 123–133.

[13] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden voice commands. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*. 513–530.

[14] Nicholas Carlini and David Wagner. 2018. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE Security and Privacy Workshops (SPW)*. IEEE, 1–7.

[15] Si Chen, Kui Ren, Sixu Piao, Cong Wang, Qian Wang, Jian Weng, Lu Su, and Aziz Mohaisen. 2017. You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 183–195.

[16] Jian Cheng, Jiaxiang Wu, Cong Leng, Yuhang Wang, and Qinghao Hu. 2017. Quantized CNN: A unified approach to accelerate and compress convolutional networks. *IEEE transactions on neural networks and learning systems* 29, 10 (2017), 4730–4743.

[17] Phillip L De Leon, Michael Pucher, Junichi Yamagishi, Inma Hernaez, and Ibon Saratxaga. 2012. Evaluation of speaker verification security and detection of HMM-based synthetic speech. *IEEE Transactions on Audio, Speech, and Language Processing* 20, 8 (2012), 2280–2290.

[18] Chunhua Deng, Siyu Liao, Yi Xie, Keshab K Parhi, Xuehai Qian, and Bo Yuan. 2018. PermDNN: Efficient compressed DNN architecture with permuted diagonal matrices. In *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 189–202.

[19] Huan Feng, Kassem Fawaz, and Kang G Shin. 2017. Continuous authentication for voice assistants. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. 343–355.

[20] Yaroslav Ganin and Victor Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*. PMLR, 1180–1189.

[21] Yang Gao, Yincheng Jin, Jagmohan Chauhan, Seokmin Choi, Jiyang Li, and Zhanpeng Jin. 2021. Voice In Ear: Spoofing-Resistant and Passphrase-Independent Body Sound Authentication. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (2021), 1–25.

[22] Yuan Gong, Jian Yang, Jacob Huber, Mitchell MacKnight, and Christian Poellabauer. 2019. ReMASC: realistic replay attack corpus for voice controlled systems. *arXiv preprint arXiv:1904.03365* (2019).

[23] Yuan Gong, Jian Yang, and Christian Poellabauer. 2020. Detecting Replay Attacks Using Multi-Channel Audio: A Neural Network-Based Method. *IEEE Signal Processing Letters* 27 (2020), 920–924.

[24] Cemal Hanilçi. 2017. Features and classifiers for replay spoofing attack detection. In *2017 10Th international conference on electrical and electronics engineering (ELECO)*. IEEE, 1187–1191.

[25] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. 2014. Deep Speech: Scaling up end-to-end speech recognition. (2014). arXiv:cs.CL/1412.5567

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[27] Yitao He, Junyu Bian, Xinyu Tong, Zihui Qian, Wei Zhu, Xiaohua Tian, and Xinbing Wang. 2019. Canceling inaudible voice commands against voice control systems. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–15.

[28] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[29] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[30] Shehzeen Hussain, Paarth Neekhara, Shlomo Dubnov, Julian McAuley, and Farinaz Koushanfar. 2021. WaveGuard: Understanding and Mitigating Audio Adversarial Examples. *arXiv preprint arXiv:2103.03344* (2021).

[31] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[32] Tomi Kinnunen, Md Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee. 2017. The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. (2017).

[33] Phillip L De Leon, Bryan Stewart, and Junichi Yamagishi. 2012. Synthetic speech discrimination using pitch pattern statistics derived from image analysis. In *Thirteenth Annual Conference of the International Speech Communication Association*.

[34] En Li, Liekang Zeng, Zhi Zhou, and Xu Chen. 2019. Edge AI: On-demand accelerating deep neural network inference via edge computing. *IEEE Transactions on Wireless Communications* 19, 1 (2019), 447–457.

[35] Zhuohang Li, Cong Shi, Yi Xie, Jian Liu, Bo Yuan, and Yingying Chen. 2020. Practical adversarial attacks against speaker recognition systems. In *Proceedings of the 21st International Workshop on Mobile Computing Systems and Applications*. 9–14.

[36] Zhuohang Li, Yi Wu, Jian Liu, Yingying Chen, and Bo Yuan. 2020. AdvPulse: Universal, Synchronization-free, and Targeted Audio Adversarial Attacks via Subsecond Perturbations. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*. 1121–1134.

[37] Meng Liu, Longbiao Wang, Jianwu Dang, Seiichi Nakagawa, Haotian Guan, and Xiangang Li. 2019. Replay attack detection using magnitude and phase information with attention-based adaptive filters. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6201–6205.

[38] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. 2018. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270* (2018).

[39] Dibya Mukhopadhyay, Maliheh Shirvanian, and Nitesh Saxena. 2015. All your voices are belong to us: Stealing voices to fool humans and machines. In *European Symposium on Research in Computer Security*. Springer, 599–621.

[40] Alan V Oppenheim, John R Buck, and Ronald W Schafer. 2001. *Discrete-time signal processing. Vol. 2.* Upper Saddle River, NJ: Prentice Hall.

[41] Nirupam Roy, Sheng Shen, Haitham Hassanieh, and Romit Roy Choudhury. 2018. Inaudible voice commands: The long-range attack and defense. In *15th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 18)*. 547–560.

[42] Md Sahidullah, Tomi Kinnunen, and Cemal Hanilçi. 2015. A comparison of features for synthetic speech detection. (2015).

[43] Jon Sanchez, Ibon Saratxaga, Inma Hernaez, Eva Navas, Daniel Erro, and Tuomo Raitio. 2015. Toward a universal synthetic speech spoofing detection using phase information. *IEEE Transactions on Information Forensics and Security* 10, 4 (2015), 810–820.

[44] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.

[45] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[46] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2018. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions. (2018). arXiv:cs.CL/1712.05884

[47] Jian Shen, Yanru Qu, Weinan Zhang, and Yong Yu. 2018. Wasserstein distance guided representation learning for domain adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[48] Cong Shi, Yan Wang, Yingying Chen, Nitesh Saxena, and Chen Wang*. 2020. WearID: Low-Effort Wearable-Assisted Authentication of Voice Commands via

Cross-Domain Comparison without Training. In *Annual Computer Security Applications Conference (ACSAC)*. 829–842.

[49] Sayaka Shiota, Fernando Villavicencio, Junichi Yamagishi, Nobutaka Ono, Isao Echizen, and Tomoko Matsui. 2015. Voice liveness detection algorithms based on pop noise caused by human breath for automatic speaker verification. In *Sixteenth annual conference of the international speech communication association*.

[50] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representation*. 1–14.

[51] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.

[52] Yu-hsin Chen, Ignacio Lopez-Moreno, Tara N Sainath, Mirkó Visontai, Raziel Alvarez, and Carolina Parada. 2015. Locally-connected and convolutional neural networks for small footprint speaker recognition. In *Proceedings of the Sixteenth Annual Conference of the International Speech Communication Association*.

[53] F Alton Everest and Ken C Pohlmann. 2015. *Master handbook of acoustics*. McGraw-Hill Education.

[54] Galina Lavrentyeva, Sergey Novoselov, Andzhukaev Tseren, Marina Volkova, Artem Gorlanov, and Alexandr Kozlov. 2019. STC antispoofing systems for the ASVspoof2019 challenge. *arXiv preprint arXiv:1904.05576* (2019).

[55] Khomdet Phapatanaburi, Longbiao Wang, Seiichi Nakagawa, and Masahiro Iwahashi. 2019. Replay attack detection using linear prediction analysis-based relative phase features. *IEEE Access* 7 (2019), 183614–183625.

[56] Mirco Ravanelli and Yoshua Bengio. 2018. Speaker recognition from raw waveform with sincnet. In *Proceedings of 2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 1021–1028.

[57] Takeshi Sugawara, Benjamin Cyr, Sara Rampazzi, Daniel Genkin, and Kevin Fu. 2020. Light commands: laser-based audio injection attacks on voice-controllable systems. In *Proceedings of the 29th {USENIX} Security Symposium ({USENIX} Security 20)*. 2631–2648.

[58] Hemlata Tak, Jose Patino, Andreas Nautsch, Nicholas Evans, and Massimiliano Todisco. 2020. Spoofing attack detection using the non-linear fusion of sub-band classifiers. *arXiv preprint arXiv:2005.10393* (2020).

[59] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. 2021. End-to-end anti-spoofing with RawNet2. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 6369–6373.

[60] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. 2019. ASVspoof 2019: Future horizons in spoofed and fake audio detection. *arXiv preprint arXiv:1904.05441* (2019).

[61] Chengyi Wang, Yu Wu, Shujie Liu, Jinyu Li, Liang Lu, Guoli Ye, and Ming Zhou. 2020. Low latency end-to-end streaming speech recognition with a scout network. *arXiv preprint arXiv:2003.10369* (2020).

[62] Xin Wang and Junich Yamagishi. 2021. A comparative study on recent neural spoofing countermeasures for synthetic speech detection. *arXiv preprint arXiv:2103.11326* (2021).

[63] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. 2018. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security* 13, 11 (2018), 2884–2896.

[64] Chen Yan, Yan Long, Xiaoyu Ji, and Wenyuan Xu. 2019. The catcher in the field: A fieldprint based spoofing detection for text-independent speaker verification. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1215–1229.

[65] Guoming Zhang, Xiaoyu Ji, Xinfeng Li, Gang Qu, and Wenyuan Xu. 2021. EarArray: Defending against DolphinAttack via Acoustic Attenuation. In *Network and Distributed Systems Security (NDSS) Symposium*.

[66] Massimiliano Todisco, Héctor Delgado, and Nicholas WD Evans. 2016. A New Feature for Automatic Speaker Verification Anti-Spoofing: Constant Q Cepstral Coefficients.. In *Odyssey*, Vol. 2016. 283–290.

[67] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7167–7176.

[68] Tavish Vaidya, Yuankai Zhang, Micah Sherr, and Clay Shields. 2015. Cocaine noodles: exploiting the gap between human and machine speech recognition. In *9th {USENIX} Workshop on Offensive Technologies ({WOOT} 15)*.

[69] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. WaveNet: A Generative Model for Raw Audio. (2016). arXiv:cs.SD/1609.03499

[70] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[71] Chen Wang, S Abhishek Anand, Jian Liu, Payton Walker, Yingying Chen, and Nitesh Saxena. 2019. Defeating hidden audio channel attacks on voice assistants via audio-induced surface vibrations. In *Proceedings of the 35th Annual Computer Security Applications Conference*. 42–56.

[72] Longbiao Wang, Yohei Yoshida, Yuta Kawakami, and Seiichi Nakagawa. 2015. Relative phase information for detecting human speech and spoofed speech. In *Sixteenth Annual Conference of the International Speech Communication Association*.

[73] Run Wang, Felix Juefei-Xu, Yihao Huang, Qing Guo, Xiaofei Xie, Lei Ma, and Yang Liu. 2020. DeepSonar: Towards Effective and Robust Detection of AI-Synthesized Fake Voices. In *Proceedings of the 28th ACM International Conference on Multimedia*. 1207–1216.

[74] Shu Wang, Jiahao Cao, Xu He, Kun Sun, and Qi Li. 2020. When the Differences in Frequency Domain are Compensated: Understanding and Defeating Modulated Replay Attacks on Automatic Speech Recognition. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (Oct 2020). https://doi.org/10.1145/3372297.3417254

[75] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. 2017. Tacotron: Towards End-to-End Speech Synthesis. (2017). arXiv:cs.CL/1703.10135

[76] Marcin Witkowski, Stanislaw Kacprzak, Piotr Zelasko, Konrad Kowalczyk, and Jakub Galka. 2017. Audio Replay Attack Detection Using High-Frequency Features. In *Interspeech*. 27–31.

[77] Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems* 2, 1-3 (1987), 37–52.

[78] Zhizheng Wu and Haizhou Li. 2013. Voice conversion and spoofing attack on speaker verification systems. In *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*. IEEE, 1–9.

[79] Xiong Xiao, Xiaohai Tian, Steven Du, Haihua Xu, Eng Siong Chng, and Haizhou Li. 2015. Spoofing speech detection using high dimensional magnitude and phase features: The NTU approach for ASVspoof 2015 challenge. In *Sixteenth Annual Conference of the International Speech Communication Association*.

[80] Ryoya Yaguchi, Sayaka Shiota, Nobutaka Ono, and Hitoshi Kiya. 2019. Replay attack detection using generalized cross-correlation of stereo signal. In *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 1–5.

[81] Hiromu Yakura and Jun Sakuma. 2018. Robust audio adversarial example for a physical attack. *arXiv preprint arXiv:1810.11793* (2018).

[82] Chao-Han Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Chin-Hui Lee. 2020. Characterizing speech adversarial examples using self-attention u-net enhancement. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3107–3111.

[83] Zhuolin Yang, Bo Li, Pin-Yu Chen, and Dawn Song. 2018. Characterizing audio adversarial examples using temporal dependency. *arXiv preprint arXiv:1809.10875* (2018).

[84] Xuejing Yuan, Yuxuan Chen, Yue Zhao, Yunhui Long, Xiaokang Liu, Kai Chen, Shengzhi Zhang, Heqing Huang, Xiaofeng Wang, and Carl A Gunter. 2018. Commandersong: A systematic approach for practical adversarial voice recognition. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*. 49–64.

[85] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. 2017. Dolphinattack: Inaudible voice commands. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 103–117.

[86] Linghan Zhang, Sheng Tan, and Jie Yang. 2017. Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 57–71.

[87] Linghan Zhang, Sheng Tan, Jie Yang, and Yingying Chen. 2016. Voicelive: A phoneme localization based liveness detection for voice authentication on smartphones. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. 1080–1091.

A APPENDIX

A.1 Microphone Configuration

Table 8: Microphone arrays in intelligent audio devices.

Device Name	# of Mics	Device Name	# of Mics
Amazon Echo (4th Gen)	6	Google Home	2
Amazon Echo (3th Gen)	7	Google Home Max	6
Amazon Echo Dot (4th Gen)	4	Google Nest Mini (2nd gen)	3
Amazon Echo Dot (3th Gen)	4	Google Nest Mini (1st gen)	2
Amazon Show 10	3	Google Nest Audio (1st gen)	3
Amazon Show 8	2	Google Nest Hub	2
Amazon Show 5	2	Google Nest Hub Max	2
Amazon Echo Studio	7	Apple HomePod	6
Amazon Echo Auto	8	Apple HomePod Mini	4

A.2 List of Speech Commands

Table 9: Speech commands used to generate attack speech samples: (a) the commands for synthesis attack, inaudible attack, and modulated replay attack, and (b) the target voice commands to generate the adversarial examples.

(a)

ID	Command
1	Please call Stella
2	Call 12345
3	Facetime 12345
4	Turn on airplane mode
5	Open the door
6	Navigation
7	Hey Siri
8	Ok Google
9	Hi Galaxy
10	Hello Huawei

(b)

ID	Command
1	Disable home alarm
2	Unlock the door
3	Browse to evil.com
4	Set volume to 0
5	Call mom
6	Power off
7	Open door
8	Call dad
9	Read email
10	Unlock iPhone

A.3 Description of the ReMASC Dataset

Table 10: Data volume of the ReMASC dataset [22].

Environment	# Subjects	# Genuine	# Replayed
Outdoor	12	960	6,900
Indoor #1	23	2,760	23,104
Indoor #2	10	1,600	7,824
In-vehicle	10	3,920	7,644
Total	55	9,240	45,472

Table 11: Data separation of the ReMASC dataset [22].

# Device	# Training	# Testing	Test Ratio
1	5,357	2,989	0.3581
2	6,126	4,538	0.4255
3	5,862	4,238	0.4196
4	6,161	4,515	0.4229