Push the Limit of Adversarial Example Attack on Speaker Recognition in Physical Domain

Qianniu Chen
Zhejiang University and ZJU-HIC*
zjcqn@zju.edu.cn

Meng Chen
Zhejiang University
meng.chen@zju.edu.cn

Li Lu*
Zhejiang University
li.lu@zju.edu.cn

Jiadi Yu
Shanghai Jiao Tong University
jiadiyu@sjtu.edu.cn

Yingying Chen
Rutgers University
yingche@scarletmail.rutgers.edu

Zhibo Wang
Zhejiang University
zhibowang@zju.edu.cn

Zhongjie Ba
Zhejiang University
zhongjieba@zju.edu.cn

Feng Lin
Zhejiang University
flin@zju.edu.cn

Kui Ren
Zhejiang University
kuiren@zju.edu.cn

ABSTRACT

The integration of deep learning on Speaker Recognition (SR) advances its development and wide deployment, but also introduces the emerging threat of adversarial examples. However, only a few existing studies investigate its practical threat in physical domain, which either evaluate its feasibility only by directly replaying generated adversarial examples, or explore the partial channel interference for robustness improvement. In this paper, we propose a physical adversarial example attack, *PhyTalker*, which could generate and inject perturbations on voices in a live-streaming manner on attacking various SR models in different physical channels. Compared with the typical adversarial example for digital attacks, *PhyTalker* generates a subphoneme-level perturbation dictionary to decouple the perturbation optimization and injection. Moreover, we introduce the channel augmentation to compensate both device and environmental distortions, as well as model ensemble to improve the perturbation transferability. Finally, *PhyTalker* recognizes and localizes the latest recorded phoneme to determine the corresponding perturbations for real-time broadcasting. Extensive experiments are conducted with a large-scale corpus in real physical scenarios, and results show that *PhyTalker* achieves an overall Attack Success Rate (ASR) of 85.5% in attacking mainstream SR systems and Mel Cepstral Distortion (MCD) of 2.45dB in human audibility.

CCS CONCEPTS

• **Security and privacy** → **Mobile and wireless security**; • **Computing methodologies** → *Artificial intelligence*;

KEYWORDS

Adversarial Example Attack; Physical Domain; Speaker Recognition; Live-streaming

ACM Reference Format:
Qianniu Chen, Meng Chen, Li Lu, Jiadi Yu, Yingying Chen, Zhibo Wang, Zhongjie Ba, Feng Lin, and Kui Ren. 2022. Push the Limit of Adversarial Example Attack on Speaker Recognition in Physical Domain. In *The 20th ACM Conference on Embedded Networked Sensor Systems (SenSys '22), November 6–9, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3560905.3568518

*Corresponding Author.

SenSys '22, November 6–9, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9886-2/22/11…$15.00
https://doi.org/10.1145/3560905.3568518

1 INTRODUCTION

Recent years have witnessed voiceprint becoming one of the most emerging biometrics, thanks to its easy integration with the natural and human-centered Voice User Interface (VUI). Benefit from the advances by deep learning, corresponding Speaker Recognition (SR) achieves wide applications on both hardware (e.g., smart speakers including Google Home Voice Match[17], Amazon Alexa Voice ID[2]) and software (mobile banks and instant messaging APPs including HSBC Bank, U.S. Bank, TD Bank [5] and WeChat [56]) systems. However, as users enjoy the convenient authentication experience of SR services, these solutions have been revealed vulnerable to adversarial example attacks, due to the intrinsic linear structure of neural networks. Such a vulnerability indicates that SR is facing severe security threats as investigated in many existing studies, and raises prevalent user privacy concerns.

Early researches [7, 26, 60] reveal the vulnerability of deep learning-based SR under white-box attacks in digital domain. Such attacks construct the adversarial examples and directly inject them to underneath machine learning models of SR systems in the digital space. To push such threats into practice, recent studies start exploring physical adversarial example attacks. The representative work FakeBob [8] evaluates its physical attack performance by directly replaying generated adversarial examples. Though its proposed query-based natural evolution strategy realizes the black-box perturbation generation, the neglect of physical perturbation injection and channel interference limit its threats to practical SR

[1]ZJU-HIC is the acronym of ZJU-Hangzhou Global Scientific and Technological Innovation Center.

systems. Instead, without using typical injection mode (i.e., digital overlaying or physical replaying), other studies [32, 59, 61] generate input-agnostic adversarial examples for real-time physical attacks. But they are still limited by poor channel robustness and weak transferability on attacking unknown models. Recent studies [34, 35, 61] take channel interference into consideration, and introduce room impulse response [50] compensation for robustness enhancement. However, they ignore the significant distortion from device channels caused by the hardware imperfection, thus remaining limited in practice. Therefore, the adversarial examples realized by all of these works only *partially* meet the demand for physical attacks, either in adversarial example injection mode or biased channel enhancement.

Toward this end, we revisit the adversarial example attack in physical domain, and comprehensively explore several key goals required to implement a practical physical attack, i.e., real-time live streaming, physical channel robustness and transferability on attacking various models. Based on the analysis, several key challenges need to be addressed to realize the physical attack. *Real-time Perturbation Generation and Injection*: to avoid being exposed to surrounding people, the attack should be performed in a live-streaming manner, requiring the real-time perturbation generation and injection physically. *Channel Interference Resistance*: the physical perturbation propagation introduces complex interference into adversarial examples, so the attack needs to be cross-channel, i.e., being able to resist both device and environmental channel distortions. *Transferability on Attacking Unknown Models*: the prior knowledge of target SR model details is hard to obtain for adversaries, indicating the demand of black-box attacking capability.

In this paper, we first investigate the system model of mainstream SR systems, and the threat model of physical adversarial example attacks. To realize the threat model, we propose *PhyTalker*, a live-streaming, cross-channel and black-box adversarial example attack on SR in physical domain. Different from typically converting an entire voice to an utterance-level adversarial example in an offline manner, we turn to generate fixed-length subphoneme-level perturbations for 40 widely-used phonemes respectively as a dictionary, which is independent of specific text context. By playing the corresponding subphoneme-level perturbations while the adversary is uttering a live voice stream, *PhyTalker* can perturb the whole utterance in real-time. Specifically, *PhyTalker* first determines the perturbation length according to the statistical analysis on a large-scale corpus, and further generates the perturbations to form the dictionary. To resist the channel interference and model variation, we introduce the channel impulse response, including both typical environmental noises and our investigated imperfect device variation, to augment the training corpus for perturbation optimization. To generalize on unknown SR systems, we adopt a model ensemble method to improve the perturbation transferability. After generating the perturbation dictionary, *PhyTalker* performs the live-streaming perturbation broadcasting for real-time injection. In particular, based on the recorded live-streaming voices of the adversary, *PhyTalker* continuously recognizes and localizes the latest recorded phoneme, and then estimates the subsequent unrecorded phonemes by referring to a preset reference phoneme sequence. With the estimated phoneme information, *PhyTalker* plays the corresponding perturbations to inject the adversary's live-streaming

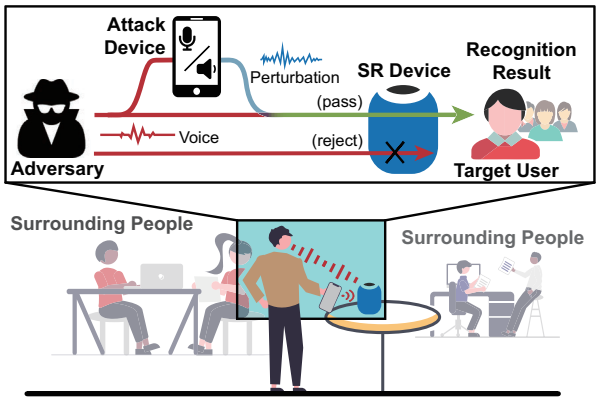


Figure 1: Threat model on speaker recognition.

voices in physical domain so as to launch an imperceptible attack. Extensive experiments in real scenarios demonstrate that *PhyTalker* can effectively deceive SR systems in physical domain while remaining imperceptible to surrounding people.

We highlight our contributions as follows.

- We explore three major requirements comprehensively underlying a practical physical adversarial example attack scenario, including live-streaming injection, cross-channel robustness, and black-box optimization.
- We propose a physical adversarial example attack, *PhyTalker*, which enables an adversary to broadcast real-time perturbations synchronously with the live voices in physical domain.
- We design a subphoneme-level adversarial perturbation generation and a perturbation-voice synchronization mechanism, which takes advantages of the composability and stability of phonemes, outperforming the universal perturbation in imperceptibility.
- We develop a channel augmentation approach and a model ensemble method, to improve perturbations' robustness on various devices and environmental channels as well as unknown SR models in physical attacks.
- We conduct extensive experiments with a large-scale corpus in real physical scenarios, and results show that *PhyTalker* can achieve an overall Attack Success Rate (ASR) of 85.5% in attacking mainstream SRs, Mel Cepstral Distortion (MCD) of 2.45dB in terms of human audibility and Real Time Factor (RTF) of 0.5 in terms of computational efficiency.

2 ATTACK STATEMENT

In this section, we present the system and threat model, then show the design goals for the physical adversarial example attack on SR, and finally present the overview of our attack.

2.1 System and Threat Models

Speaker Recognition (SR) is an automatic technique that extracts distinguishable voiceprint from raw voices to identify speakers. And its rapid development has derived many categories, including text-dependent [29] and text-independent recognition [6], Open-Set Identification (OSI) [15], Close-Set Identification (CSI) [36]

and Speaker Verification (SV) [13], etc. Considering the generalization capability and adaptability, our attack targets on the text-independent OSI SR system. Specifically, such a system extracts the high-level embedding features underlying speaker voices, then derives the confidence corresponding to the enrolled user profiles, and finally regards the user identity as the one with the highest confidence. To further prevent the spoofers from accessing, the system further sets a confidence threshold to reject speakers with lower confidence. Mathematically, the decision of an OSI system can be formulated as:

$$D(x) = \begin{cases} \arg\max_{y_i}(S_{y_i}(x)), & \max_{i\in\{0,\cdots,n-1\}} S_i(x) > \theta \\ reject, & otherwise, \end{cases} \quad (1)$$

where x is the testing voice, $S_{y_i}(x)$ is the confidence that the voice x belongs to the enrolled user y_i, and θ denotes the confidence threshold.

To realize the aforementioned physical attack on SR system, we need to achieve the following design goals.

To physically attack such an SR system, we then illustrated our threat model as shown in Figure 1. In this attack, an adversary intends to impersonate a legitimate user to access the target SR system, so as to retrieve the user's privacy or activate sensitive voice commands. We assume the adversary has not enrolled in the target system, thus being regarded as a spoofer in normal situations. To obtain legitimate access, a straightforward approach is to replay the victim's voice or a pre-crafted forged voice directly. But such an approach is limited to the scenarios in the absence of other persons, because the broadcasting voice through common COTS loudspeakers induces a significantly different hearing from live speech voice, thus raising ambient persons' awareness and leading to the failure of an imperceptible attack. To this end, our threat model is considered to be able to launch the imperceptible attack on this extended scenario, where other persons are allowed to exist around the adversary and target SR system. For example, an adversary wants to impersonate a legitimate user to punch in the voiceprint attendance system that is fixed in an office with surrounding colleagues. Since the colleagues around may see/hear the attendance system and the adversary, the adversary cannot replay the legitimate user's voices only without live speech, which easily attracts surrounding persons' awareness. On the other hand, we assume the adversary can collect a few voice examples longer than 5s of the target user from public conversations or social media. Representative examples include crank calls, videos from TicTok or Youtube, etc. But note that the content of the collected voice sample is not constrained thus not required to cover the texts that may be used in further attacks. Considering the limited texts in collected samples and attention-attracted scenarios, the adversary obviously cannot launch such an attack via pure replay techniques. Also, we assume the adversary has no prior knowledge of the target SR system, including signal processing techniques, neural network structure, model parameter configuration, etc. And the adversary could neither control the receiver device, nor the environmental settings of the target SR system. Furthermore, the adversary is only allowed to carry an attack device with a speaker and a microphone, which is used to broadcast the adversarial perturbations.

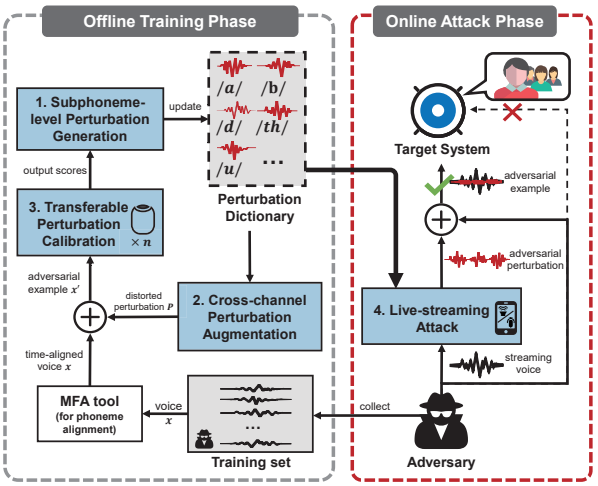


Figure 2: Overview of *PhyTalker*.

2.2 Design Goals

Live-streaming. As mentioned in Section 2.1, the adversary can not simply replay the user's voice or adversarial examples without live speech in order to avoid raising surrounding persons' awareness. Hence, the adversary needs to utter live-streaming voices and inject adversarial perturbations in real-time. This introduces the requirements of real-time perturbation generation and voice-perturbation synchronization, indicating more challenges compared with digital-domain attacks.

Cross-channel. Unlike over-the-line digital attacks, the generated adversarial examples experience microphone transmission, over-the-air propagation and loudspeaker reception in physical domain. The complex channel, including various device models and different surrounding environments, induces severe distortions to the audio signals. To ensure that the adversarial perturbations survive in the target physical domain, it is necessary to enhance their cross-channel robustness.

Black-box. Commercial SR systems usually do not expose their model details to the public. Even worse, most systems limit the query frequency to prevent enumeration-like attacks. Hence, for an adversarial example attack, especially the physical attack, the adversary has no prior knowledge of the SR system. This requires the adversary to realize a black-box attack in real situations.

2.3 Attack Overview

Meeting the goals mentioned in Section 2.2, we propose *PhyTalker*, a live-streaming, cross-channel and black-box attack. The basic idea of *PhyTalker* is to decouple the perturbation optimization and injection by generating a dictionary of subphoneme-level, cross-channel and transferable adversarial perturbations for live-streaming attacks. Using the dictionary, the adversary launchs a stealthy physical adversarial example attack by uttering live-streaming voices and broadcasting imperceptible adversarial perturbations simultaneously. Figure 2 shows the overview of *PhyTalker*, including the offline and online phases.

The offline phase aims to generate a dictionary of cross-channel and transferable subphoneme-level perturbations for live-streaming attacks. First, in *Subphoneme-level Perturbation Generation*, we generate the subphoneme-level perturbations with a fixed length for each phoneme to construct the dictionary in the offline attack preparation. Specifically, *PhyTalker* extracts the phoneme sequence from the training voices, and aligns voice and the corresponding phoneme based on timestamp with a Kaldi-based open-source tool Montreal Forced Aligner (MFA). Then *PhyTalker* iteratively optimizes the perturbation for each phoneme by solving the well-designed objective function, so as to generate an adversarial perturbation dictionary. During the optimization, *Cross-channel Perturbation Augmentation* employs a channel simulation method with Unit Impulse Response(UIR) to augment adversarial perturbations. By simulating various channels from different device models and environments with only a few samples, *PhyTalker* could adapt to different channel interference in physical attacks. Furthermore, we ensemble multiple mature SR models for the perturbation optimization in *Transferable Perturbation Calibration* to support the black-box attacking capability.

The online phase aims to broadcast adversarial perturbation with the well-trained subphoneme-level perturbation dictionary for the physical perturbation injection in real-time. In particular, in *Live-streaming Adversarial Example Attack*, *PhyTalker* first constructs a Reference Phoneme Sequence (RPS) for real-time phoneme alignment. With a well-trained recurrent neural network model, *PhyTalker* recognizes and localize the latest recorded phoneme and predict its subsequent phonemes relative to the reperfence phoneme sequence. Then, *PhyTalker* employs an Exponentially Weighted Moving-Average (EWMA) algorithm to estimate the time duration of subsequent phonemes. With the estimated phoneme information, *PhyTalker* could broadcast subphoneme-level adversarial perturbations of the appropriate type and number. In the over-the-air propagation, the adversary's live-streaming voice would be injected with adversarial perturbations, and then spoof the target SR system to regard it as from a legitimate user.

3 ATTACK DESIGN

In this section, we present the design detail of *PhyTalker*.

3.1 Subphoneme-level Adversarial Perturbation Generation

As mentioned in Section 2.2, our attack aims to be live-streaming so as to avoid raising the awareness of surrounding persons. Hence, *PhyTalker* needs to generate the adversarial perturbations corresponding to the adversary's speech in real-time. However, typical adversarial example attacks usually require the whole adversary's speech voice (i.e., an utterance-level voice) as input for the perturbation training, indicating that the generated perturbations highly depend on the input voice texts. In a physical attack, the adversary probably needs to speak various commands to achieve his/her curious or malicious objectives. Hence, such an implementation cannot meet the live-streaming goal in the physical attack. To realize the live-streaming attack, we decouple the perturbation generation and injection processes, and propose to generate a subphoneme-level adversarial perturbation dictionary in the offline generation for further online injection.

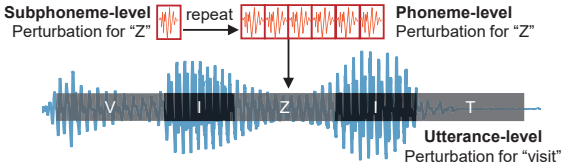


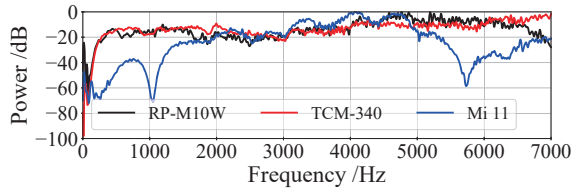
Figure 3: Illustration of adversarial example generation with subphoneme-level perturbation, where the subphoneme-level perturbation repeats to form phoneme-level perturbation, then forms utterance-level perturbation.

Phonemes are the basic units of voice in phonology. A limited number of phonemes can be combined into any semantic words and sentences in a language, e.g., English only has 39 widely-used phonemes, as shown in Table 1. Also, these basic phonemes have relatively stable vocal characteristics, such as similar frequency distribution and formants, indicating that phoneme-based features tend to be similar among utterances with different text or contexts. Inspired by the composability and stability of phonemes, we generate a perturbation dictionary where each perturbation corresponds to each phoneme, and employ it to construct adversarial examples with any text. Unfortunately, due to the variation of speech texts, context and speed, the duration of phonemes is not fixed, even the same phoneme in different contexts (e.g., different words) with distinct duration. To handle it, we propose to generate fine-grained subphoneme-level perturbations, whose duration is fixed and shorter than that of all phonemes.

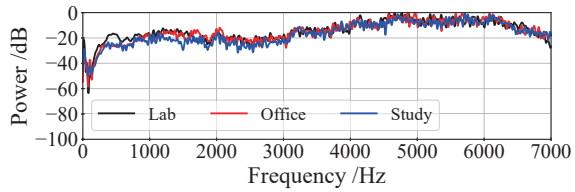
Specifically, we define $P = \{P_1, P_2, ..., P_{40}\}$ as a dictionary of generated subphoneme-level perturbations, consisting of 40 perturbations for 39 phonemes in Table 1 and the silence interval. For a given input voice x, we decompose it into a phoneme sequence and inject the corresponding perturbations for each phoneme. Note that the subphoneme-level perturbation is much shorter than the phoneme duration, we repeat the same perturbation as many as

Table 1: List of ARPAbet [25] phonemes.

No.	Phn	Eg.	No.	Phn	Eg.	No.	Phn	Eg.
1	AA	odd	14	F	fee	27	P	pee
2	AE	at	15	G	green	28	R	read
3	AH	hut	16	HH	he	29	S	sea
4	AO	ought	17	IH	it	30	SH	she
5	AW	cow	18	IY	eat	31	T	tea
6	AY	hide	19	JH	gee	32	TH	theta
7	B	be	20	K	key	33	UH	hood
8	CH	cheese	21	L	lee	34	UW	two
9	D	dee	22	M	me	35	V	vee
10	DH	thee	23	N	knee	36	W	we
11	EH	Ed	24	NG	ping	37	Y	yield
12	ER	hurt	25	OW	oat	38	Z	zee
13	EY	ate	26	OY	toy	39	ZH	seizure



(a) Three different device models



(b) Three different environments

Figure 4: Frequency responses of loudspeaker-environment-microphone systems under different device models and environments.

possible within each phoneme duration. With such a subphoneme-level perturbation injection, we can adapt to different phonemes of variant length and then generate the adversarial example $x^{'}$. Figure 3 illustrates an example of perturbation injection for word "visit".

To realize a successful targeted attack, the generated adversarial example $x^{'}$ should reach the maximum confidence for the target speaker y_t while larger than the preset threshold θ. Hence, our objective function is designed as follows:

$$L_{S,\theta}(x^{'}, y_t) = \max\{\theta, \max_{i \in Y \setminus \{y_t\}} S_i(x^{'})\} - S_{y_t}(x^{'}). \tag{2}$$

To generate P for phonemes from general voices with any text, we further employ batch of voice samples following a distribution $\mathcal{X}$, instead of a single sample, for perturbation optimization:

$$\begin{aligned} \arg\min_{P} \quad & \mathbb{E}_{x \sim \mathcal{X}} L_{S,\theta}(G(x, P), y_t) \\ \text{s.t.} \quad & \|P\|_\infty \leq \epsilon, \end{aligned} \tag{3}$$

where $G(x, P)$ denotes the perturbation generation function from a voice sample x based on the dictionary P, i.e., $x^{'} = G(x, P)$, ϵ refers to the perturbation scale.

By solving Eq. (3), we obtain a subphoneme-level perturbation dictionary, with which *PhyTalker* generates utterance-level adversarial examples with various texts by injecting subphoneme-level perturbations on each phoneme to launch the impersonation attack.

3.2 Cross-channel Adversarial Perturbation Augmentation

Though the subphoneme-level perturbation dictionary contributes to a successful attack in digital domain, it may fail due to complex physical propagation channels, as discussed in Section 2.2. To improve the robustness of *PhyTalker* in physical domain, we adopt a channel simulation method for low-effort channel augmentation on the adversarial examples.

Theoretically, when broadcasting over the air, the digital perturbation first travels through DAC circuit, voice coil and diaphragm, becoming mechanical acoustic waves, then propagates through the air medium in a space, and finally received by a microphone, experiencing ADC and other electronic circuits to generate the digital signals. Such a transmission process introduces two aspects of channel interference, i.e., device-related interference (e.g., non-linearity of the amplifier) and space-related interference (e.g., multi-path effect).

To explicitly observe channel interference on voices, we measure frequency responses of different devices and environments. On one hand, we employ RP-M10W, TCM-340, Mi11 as receiver respectively in the same environment (i.e., a lab) to study the impact of different devices. On the other hand, we employ RPM10W as the same receiver in a lab, an office, a study respectively to explore the environmental variations. Considering the transmitter device can be controlled by the adversary, we use the same loudspeaker EDIFIER M230, and fix the loudspeaker and microphone at a distance of 5cm. Figure 4 shows the frequency responses under different device models and environments. We can observe that different device models exhibit significant differences between each other in frequency responses, while different environments also induce considerable variants into the received signals. The result indicates that the channels, including the device and environment, introduces distortion into the over-the-air signals. Considering the subtle perturbations, such interference significantly downgrades the performance of physical adversarial example attacks.

Based on the analysis, *PhyTalker* augments the perturbation dictionary P with various UIR. Typically, the most straightforward augmentation approach is to collect a large amount of perturbation broadcasting samples using different device models in different environments. However, such an approach requires significant data collection efforts, which limits the threat of physical attacks. To release the adversary's efforts, we design a channel simulation method for perturbation augmentation. The basic idea is to decouple the received signal into pure voice and channel response. Theoretically, a signal x propagating through a channel c becomes $y = c * x$. In practice, the digital signal is the one involved in channel responses, i.e., y in the equation,instead of x. This indicates that we could simulate a signal propagating through any channel in digital domain as long as the channel response is obtained, providing us the opportunity to augment the perturbation for improving its channel robustness.

To derive the channel response, we consider the whole system propagated by the voice as a Linear, Time-Invariant (LTI) system, because of the relatively wide linear regions offered by most audio systems. Fortunately, for a LTI system, the channel response can be formulated by Unit Impulse Response (UIR). To measure the UIR, we employ MLS-derived impulse response measurements[46], one of the basic acoustics measurement methods in ISO standards[22]. Specifically, we broadcast an MLS signal (i.e., a specific kind of pseudo-random binary signal) with a transmitter, then propagating through a specific environment, and finally received by a receiver. After that, the channel's UIR, including the transmitter-receiver device models and environment, is derived by performing convolution operation on the received signal and transmitted MLS signal. Since the MLS signal is spectrally flat, pre-certain and repeatable,

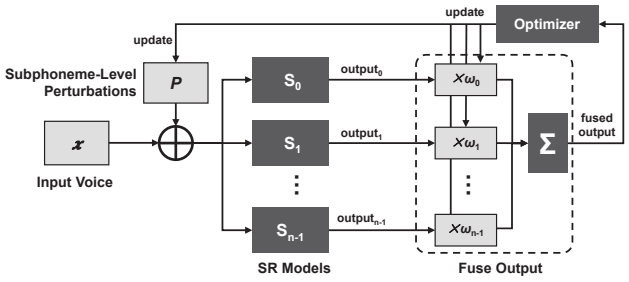


Figure 5: Illustration of perturbation optimization with ensemble learning and adaptive weights.

the derived UIR has a higher Signal-to-Noise Ratio (SNR). Using such a method, the adversary could collect only one sample in each environment under each device model to generate various channel responses.

After obtaining various UIRs, we augment the utterance-level adversarial perturbation via a convolution operation before injecting it into the voice. Then, our perturbation optimization problem Eq. (3) is transformed as:

$$\begin{aligned} \arg\min_{P} \quad & \mathop{\mathbb{E}}_{x\sim X, c\sim C} L_{S,\theta}(G(x, P * c), y_t) \\ \text{s.t.} \quad & \|P\|_\infty \le \epsilon, \end{aligned} \tag{4}$$

where C is a UIR distribution collected from multiple real transmitter-receiver devices and environments, excluding the targeted devices and environment.

By augmenting subphoneme-level perturbations with collected UIRs, *PhyTalker* can cross various device models and different environments to launch a successful adversarial example attack in physical domain without the knowledge of the victim's device and environment.

3.3 Transferable Adversarial Perturbation Calibration

Except for the channel robustness, it's also necessary to realize the black-box attack on SR systems, as mentioned in Section 2.1. Hence, *PhyTalker* needs to improve its transferability so that the attack still remains successful when meeting unknown systems.

To improve *PhyTalker*'s transferability, we employ the ensemble learning method to fuse the outputs from multiple substitute SR models in the perturbation optimization. Specifically, when a voice sample x is fed to the optimization process, the corresponding adversarial perturbation is generated from P as mentioned in Section 3.1. Then, the generated adversarial example is fed to n different substitute SR models, $(S_0, \theta_0), ..., (S_{n-1}, \theta_{n-1})$, to obtain various confidence outputs, which are fused to update perturbations for discarding model-specific features and learning transferable perturbations. Considering the distinct importance of different SR models, we further introduce learnable adaptive weight with weight normalization, $w_i \in \{w_0, ..., w_{n-1}\}$ where $\sum_{i=0}^{n-1} w_i = 1$, for each model to dynamically adjust the significance of each model as shown in Figure 5. Hence, the perturbation optimization problem is further transformed as:

$$\begin{aligned} \arg\min_{P} \quad & \mathop{\mathbb{E}}_{x\sim X, c\sim C} \sum_{i=0}^{n-1} w_i L_{S_i,\theta_i}(G(x, P * c), y_t) \\ \text{s.t.} \quad & \|P\|_\infty \le \epsilon. \end{aligned} \tag{5}$$

With ensemble learning and adaptive weights, the calibrated perturbation extends its attack targets from a single SR system to various systems, realizing the black-box attack ability.

3.4 Live-streaming Adversarial Example Attack

After the offline perturbation optimization, *PhyTalker* generates subphoneme-level, cross-channel and transferable adversarial perturbations. In the online attacking, *PhyTalker* injects the perturbations to the adversary's live-streaming voices in real-time and continuously, which consists of the real-time alignment and phoneme sequence estimation.

3.4.1 Real-time Alignment. The live-streaming attack relies on a continuous and real-time alignment between the latest recorded live-streaming voice and speech texts. After that, we estimates the types and durations of phonemes in the subsequent live-streaming voice for perturbation injection. Although the adversary's speech texts are not constrained, he/she should determine the speech texts before launching the physical attack. Thus, *PhyTalker* is able to obtain the texts of the adversary's voice before the live-streaming perturbation broadcasting. From the speech texts, we construct a Reference Phoneme Sequence (RPS), consisting of phonemes used in the live-streaming query, serving as the alignment reference. Then we use a fast phoneme recognition system to extract phoneme sequence from the latest recorded live-streaming voice and align it to RPS for perturbation synchronization.

The premises of the real-time alignment are to derive the RPS from the speech text and extract recorded phoneme sequence from the latest recorded live-streaming voice. Hence, we employ a widely-used text-to-phoneme tool (i.e., Phonemizer[4]) to extract RPS from the speech text before attacks. For each phoneme in RPS, we assign the average duration of this type of phonemes to it as duration reference, thus we finally obtain an RPS with both phoneme types and durations. Different from RPS, the recorded sequence should be regularly extracted from the latest recorded voice in real-time when the attack performs. Hence, we employ a lightweight three-layer Bidirectional Recurrent Neural Network (BRNN)[19] model for the frame-wise real-time phoneme recognition. The BRNN takes 26-dimensional acoustic features as input, and adopts 256 one-cell memory GRU units with layer normalization in both the forward and backward layers, resulting in 40-dimensional phoneme predictions. After that, we further calibrate it by combining the same or similar phonemes and discarding the incorrect or too short ones. Finally, we derive the time-aligned phoneme sequence in the recorded voice.

With RPS and the phoneme recognition system, *PhyTalker* finally recognizes the phoneme sequence in the recorded live-streaming voice and align it to the RPS with a long short-term window mechanism as shown in Figure 6. The long-term window determines a global searching interval to performance coarse-grained alignment, which starts from the last aligned phoneme with a preset window length of L. Then a short-term window is used to locate

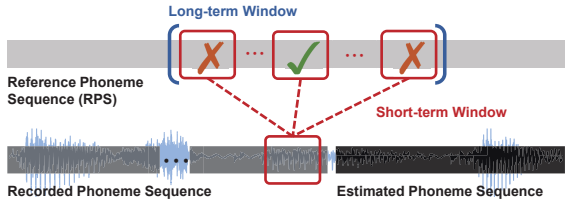


Figure 6: Illustration of the live-streaming phoneme alignment with long short-term window mechanism.

the latest recorded phoneme more accurately for fine-grained alignment. In particular, a short-term window slides through the whole long-term window in RPS, and applies Levenshtein distance[30] to measure its similarity with the one covering the latest phoneme in the recorded sequence. When the minimum Levenshtein distance is derived, *PhyTalker* regards the last phoneme in the corresponding short-term window as the aligned position in the query. We also preset a distance threshold to discard false alignment. During real-time alignment, We pull the short-term window in the recorded sequence backward phoneme-by-phoneme, and repeat the above processes. Due to the variable speaking speed of adversaries, the aforementioned alignment is continuously repeated in real-time so as to avoid the cumulative errors in the alignment.

3.4.2 Phoneme Sequence Estimation. After aligning the recorded phoneme sequence and the RPS, we can determine the phoneme types of the estimated phoneme sequence by checking the phoneme sequence after the alignment position in the RPS. To accurately inject perturbation on the subsequent live-streaming voice, the durations of each phoneme in the estimated sequence are also required. However, such phoneme durations remain unknown in the estimated sequence, and are hard to predict based on RPS due to the variable speaking speed of humans. Hence, we design a phoneme duration estimation method based on Exponentially Weighted Moving-Average (EWMA) algorithm, which dynamically adjusts the duration of phoneme-level adversarial perturbation to follow the live-streaming voice.

The basic idea of the duration estimation is to normalize the voice speed from different speeches into the same scale of RPS. Specifically, we derive a speaking speed v_i for the i^{th} phoneme relative to the speed of RPS, i.e.,

$$v_i = \frac{d_i^{ref}}{d_i^{rec}}, \tag{6}$$

where d_i^{ref} is the i^{th} phoneme's duration in RPS, d_i^{rec} is the corresponding phoneme's duration in the recorded voice. To estimate the speed of phoneme v_i^e, we derive a cumulative speed among the previous k phonemes based on EWMA, i.e.,

$$v_{i-j}^e = \beta v_{i-j} + (1-\beta) v_{i-j-1}^e, j \in \{1, \cdots, k\}, \tag{7}$$

where β is a preset weight. Considering the human speaking behavior is approximately a LTI system , the speed should lean to stable for a period of time. Hence, based on the estimated speed, *PhyTalker* further re-scales the phoneme durations in RPS to that in the recorded voices, i.e.,

$$d_{i+j}^e = \frac{d_{i+j}^{ref}}{v_{i-1}^e}, j \in \{0, \cdots, M\}, \tag{8}$$

where M is the segment size for phoneme duration estimation.

After that, we can estimate the duration of subsequent phonemes until the next alignment, and determine the number of corresponding subphoneme-level perturbations for injection. In particular, we assume the duration of each generated subphoneme-level perturbation is d_i^s (usually shorter than the shortest phoneme as mentioned in Section 3.1). When each subsequent phoneme is predicted as p with the duration of d_p^e based on Eq. (8), *PhyTalker* repeats the corresponding subphoneme-level perturbation $\lfloor d_p^e / d_p^s \rfloor$ times as the phoneme-level perturbation, and then broadcasts it by speakers for injection.

4 EVALUATION

In this section, we evaluate the performance of *PhyTalker* with open datasets in real physical environments.

4.1 Experimental Setup & Methodology

Dataset. We employ two open datasets, i.e., *VoxCeleb1*[41] and *LibriSpeech*[44], for SR model training and attack evaluation, respectively. Considering the multiple SR systems required to implement in the evaluation, we randomly select three subsets from VoxCeleb1 as the training sets for SR systems, named Train-1, Train-2, Train-3. Each subset contains 1,000 speakers with around 175k utterances. Each SR system is enrolled by 5 speakers from LibriSpeech's train-clean-100 set, including 3 males and 2 females with 5 randomly selected utterances for each speaker. On the other hand, we randomly select 10 speakers with 1,105 utterances from LibriSpeech as adversaries for the evaluation, among which 884 utterances are used for perturbation optimization while the rest 221 for testing.

SR Models. We implement 9 SR models with 3 different architectures and 3 different training datasets on PyTorch for perturbation optimization and evaluation. For all the models, we use 24-dimension MFCC features, whose frame size and step are 25ms and 10ms, respectively. And the models output 512-dimension identity embeddings. After that, a GPLDA is employed to compare the similarity between the identity embeddings of the input voice and that of each enrolled user, outputting a confidence. Table 2 shows the details and performances of the 9 SR models.

Attack Implementation. We implement the subphoneme-level perturbation generation on an AMAX server (Intel Xeon Silver 4210R, 256GB RAM, NVIDIA RTX A6000) for the attack. The dictionary of subphoneme-level perturbations are generated by minimizing the objective function in Eq. (5). By default, the amplitude

Table 2: EERs of the 9 SR models with different architectures trained by different datasets.

Architecture	Train-Set 1	Train-Set 2	Train-Set 3
x-vecter[49]	A (6.0%)	B (6.3%)	C (6.4%)
d-vecter[53]	D (9.6%)	E (10.4%)	F (9.9%)
DeepSpeaker[31]	G (7.2%)	H (8.4%)	I (7.9%)

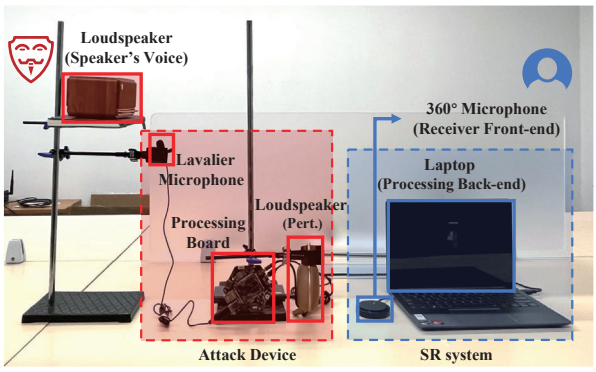


Figure 7: Illustration of experimental setup.

threshold $\epsilon = 0.02$, and the duration of subphoneme-level perturbation is 12.5ms. We employ 24 UIRs measured by 8 different recording device models with each playback device in 3 different environments for data augmentation. Note that the perturbations are specifically augmented for each single playback device, because an adversary should only use a single device for one attack. We further employ 4 white-box models with different architectures and training datasets, selected from the aforementioned 9 SR models, for the model ensemble.

Experimental Setup. Figure 7 shows the experimental setup for the evaluation. We deploy *PhyTalker* on a Seeed ReSpeaker Core v2[48] (Cortex A7, 1GB RAM) as the adversary's attack device. To control the experimental variable, we playback human voices with a loudspeaker EDIFIER M230, instead of recruiting humans for simplicity. And the target SR model is deployed on a Lenovo Xiaoxin Pro 13 with an external receiving front-end (i.e., a 360° microphone RunPu M10W, a lavalier microphone Takstar TCM-340). In each experiment, we play the voices with M230 as the adversary's real-time query. During the voice playing, the attack device receives them by a lavalier microphone, which is fixed near M230, and then derives the corresponding perturbations based on the signals, as mentioned in Section 3.4. After that, the perturbations are broadcasted with two loudspeakers (i.e., JBL CLIP 3 and HP DHS) respectively, for injection on the voice query. To simulate a real physical attack, the M230 for playing adversary's query is 50 cm (40cm × 30cm) away from the SR system's front-end, while another speaker (i.e., JBL CLIP 3 or HP DHS) is 5cm. In order to eliminate the accumulated error, the alignment for the perturbation on human voices is performed every 0.5s for 3s recorded segment. Moreover, we set a delay of 0.2 ∼ 0.4s to compensate for the time cost of data processing (around 0.17 ∼ 0.23s, measured from our implementation) and signal transmission (a random value obeying the uniform distribution $U(0.05, 0.1)$). The experiments are repeated in four different indoor environments: a Lab ($7.4 \times 5.6\text{m}^2$, 38.7dBA), an Office ($18.0 \times 6.0\text{m}^2$, 43.1 dBA), a Study ($3.1 \times 4.4\text{m}^2$, 37.2 dBA) and a Cafe simulated with acoustic scene records from TUT2016[40] in the Lab ($7.4 \times 5.6\text{m}^2$, 54.3dBA).

Metrics. (1) *Attack Success Rate (ASR)* is the ratio of successful attacks n_s among all the attack attempts n_a, i.e., $ASR = n_s/n_a \times 100\%$. (2) *Sound Pressure Level (SPL)* is a logarithmic measure of the effective pressure of a sound p relative to a reference value p_{ref}, i.e., $SPL = 20 \log p/p_{ref}$, where $p_{ref} = 20\mu Pa$ (a common value in most cases). (3) *Hit Rate (HR)* is the ratio of correct phonemes that the perturbation hits among all phonemes in an utterance. Note that a valid hit is counted only when the phoneme overlapping is over 12.5ms to contain at least one subphoneme-level perturbation. (4) *Mel Cepstral Distortion (MCD)* [27] measures the sound quality by comparing the distance between the target and reference sounds, i.e., $MCD = (10/T \ln 10) \sum_{t=1}^{T} \sqrt{2 \cdot \sum_{i=1}^{24} (mc_i^t - mc_i^e)^2}$, where mc_i^t and mc_i^e denote the target and estimated mel-cepstrals, respectively. (5) *Real Time Factor (RTF)* is the ratio of total processing time t_p to the input audio duration t_t, i.e., $RTF = t_p/t_t$. (6) *Signal-to-Noise Ratio (SNR)* is the ratio of the power of a signal P_s to the power of noise P_n expressed in decibels, i.e., $SNR = 10 \times log_{10}(P_s/P_n)$.

4.2 Overall Performance

We first evaluate the overall performance of *PhyTalker*, whose experiment is performed in the lab with CLIP3 and M10W as the front-ends of the attack device and SR system, respectively. The target SR models are A, D and G, as mentioned in Table 2. To evaluate its black-box attacking capability, we ensemble (E, F, H, I), (B, C, H, I) and (B, C, E, F) to attack model A, D, G, respectively. To validate the effectiveness of *PhyTalker*, we introduce two State-Of-The-Art (SOTA) works *FakeBob*[8] and *AdvPulse*[35] as baselines. Specifically, we implement *FakeBob* based on its open-source code [1], where we set the amplitude threshold ϵ as 0.05 and the score threshold κ as 70.0, then playback the well-crafted examples at a loud volume 50cm away from the SR system. As for *AdvPulse*, due to the lack of the open-source code, we directly refer to its available results under a similar experimental setup, for a fair comparison.

Table 3 shows ASRs, SNRs, and RTFs of *PhyTalker*, *FakeBob* and *AdvPulse* when attacking different SR systems with d-vector, x-vector and DeepSpeaker. We can find that when attacking the three models physically, *PhyTalker* all achieves 15.5% higher ASRs compared with *FakeBob*[8] on average while realizing a 5.2dB higher SNR. This indicates that *PhyTalker* outperforms *FakeBob* under the over-the-air and black-box settings. In addition, we measure MCDs of signals received by a microphone 1m away from the target SR system to evaluate the perceptibility of surrounding people. It can be observed that MCD of *PhyTalker* is 1.7dB lower than *FakeBob*[8], indicating a better audibility of *PhyTalker*. This is because *PhyTalker* decouples the perturbation with the adversary's voice. By placing the attack device near the target SR system, the volume of perturbation could be set at a relatively small value, thus leading to a lower MCD in the surrounding. Compared with the white-box *AdvPulse* on x-vector, although *PhyTalker* is a bit lower on ASR by 9.4%, its SNR is 12.1 dB better than AdvPulse, indicating that

Table 3: Overall ASRs, SNR, MCD and RTF of *PhyTalker* and SOTA works under physical attack scenarios.

Attack	ASR(%)			SNR	MCD	RTF
	d-vec.	x-vec.	D.S.	(dB)	(dB)	
PhyTalker	85.5	80.5	90.5	16.8	2.45	0.5
FakeBob	63.3	77.4	69.8	11.6	4.15	95.3
AdvPulse	N/A	89.9	N/A	4.7	N/A	<1.0

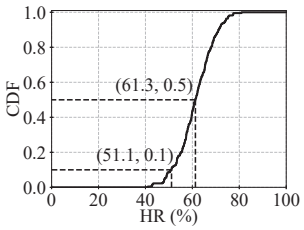


Figure 8: CDF of HRs.

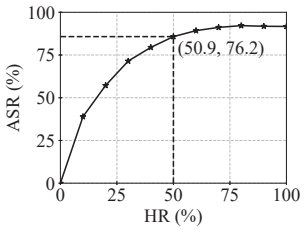


Figure 9: ASRs of *PhyTalker* with different HRs.

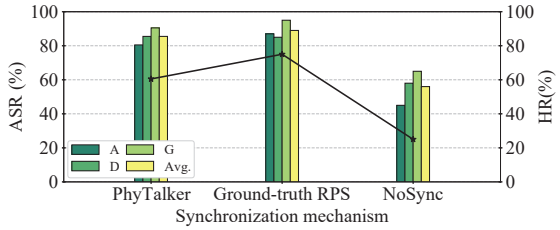
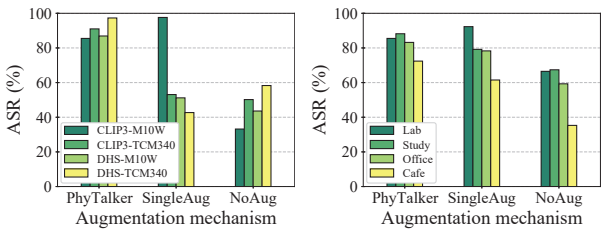


Figure 10: ASRs and HRs of *PhyTalker* and another two mechanisms.



(a) Different device models (b) Different environments

Figure 11: ASRs of *PhyTalker* under different device models and environments.

the phoneme-based optimization improves the audibility of live-streaming attacks, which outperforms SOTA significantly under a comparable attack ability.

We also compare the efficiency of generating adversarial examples. We can see from Table 3 that the average RTF of *PhyTalker* and *FakeBob* are 0.4 and 95.3, respectively. Such a significant gap is because *PhyTalker* decouples the perturbation optimization and injection, leading to only once perturbation optimization. In the process of generating adversarial examples, it is only necessary to extract phonemes in voice in real time and play the well-crafted subphoneme-level perturbations. But in *FakeBob*, the generation of adversarial examples requires a large number of queries to obtain the scores (e.g., 20k queries totally per sample in this experiment), thus increasing its time cost. Both of *PhyTalker* and *Advpulse* generate adversarial examples in real time. That's because both of them adopt the manner of online injecting well-trained adversarial perturbations thus saving the time of training perturbation during the attack.

4.3 Performance of Live-Streaming Attack

As a physical attack, *PhyTalker* enables the adversary to query in a live-streaming manner, while broadcasting corresponding perturbations in real-time. Hence, its performance highly depends on the accuracy and efficiency of synchronization between perturbations and voices as mentioned in Section 3.4.

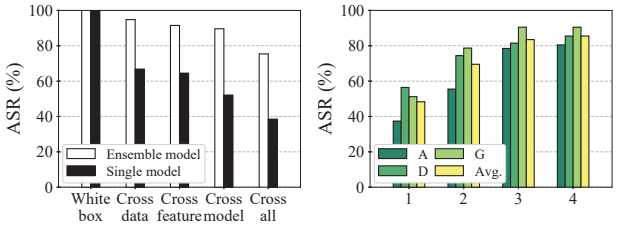
Figure 8 shows the Cumulative Distribution Function (CDF) of HR after applying our synchronization strategy. We can observe that the HR mainly ranges from 40% to 80% with a median value of 61.2% and over 90% of utterances achieve over 50% HR. This indicates that *PhyTalker* successfully perturbs at least 50% phonemes in any live-streaming voice. We further investigate the impact of synchronization performance on the attack effectiveness. Figure 9 shows ASRs of *PhyTalker* over different HRs. As the HR grows, ASR rapidly increases and then gradually go stable to 91% ASR. Note that the ASR has already reached 76.2% at the point of 50% HR, and can still be improved with higher HR, indicating the satisfactory synchronization performance for live-streaming attack. In addition, the synchronization efficiency is also important for real-time perturbation injection. Hence, we also evaluate the time complexity of our synchronization strategy. The result derived from all the testing samples shows that the time cost ranges from 0.17s~0.23s with a mean value of 0.21s, which is much less than the synchronization step (0.5s) and sufficient for real-time injection.

We also evaluate *PhyTalker* under different synchronization mechanisms. Except for the proposed mechanism in Section 3.4, we implement another two mechanisms, i.e., (1) *Ground-truth RPS*: using the ground-truth phoneme sequence and duration as RPS, (2) *NoSync*: directly broadcast perturbation according to the RPS starting with the voice without synchronization. Figure 10 shows the HRs and ASRs of using *PhyTalker* and another two mechanisms. Compared with *NoSync*, both the average ASR and HR of *PhyTalker* are higher with 29.7% and 35.1%, respectively. This result indicates that the regular synchronization mechanism of *PhyTalker* plays an important role in enhancing the attack performance in physical domain. On the other hand, we can find the HR of *PhyTalker* is 21.8% lower than that of *Ground-truth RPS*, but the ASR only declines 2.4%. This is because even though a phoneme is not accurately estimated, the corresponding perturbation could be still injected near the phoneme due to the regular alignment. This result further supports a well-performed synchronization in *PhyTalker*.

4.4 Performance of Channel Robustness

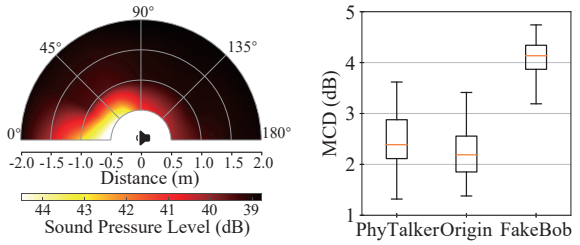
The robustness of a physical attack to various channel interference is another important property. We evaluate the performance of *PhyTalker* under different channels of device models and environments. In the experiment, we implement two other augmentation mechanisms as baselines, i.e., (1) *SingleAug* applies only target device model pair (i.e., CLIP3-M10W) and one environment (i.e., lab) for augmentation. (2) *NoAug* generates the perturbation without augmentation.

Figure 11(a) shows ASRs of *PhyTalker* under four different device model pairs (i.e., CLIP3 and DHS as loudspeaker, M10W and TCM340 as receiver). We can see that *PhyTalker* achieves ASRs of 85.5%, 91.0%, 86.9% and 97.3% on the four device models, respectively. The ASRs under M10W as the receiver are slightly lower, because M10W is a conference microphone with higher sensitivity



(a) Type of target models (b) Number of models for ensemble

Figure 12: ASRs of *PhyTalker* under different models.



(a) SPL distribution of the adversarial perturbation under the ambient noise of 38.7dB

(b) MCDs of re-recorded original voices and adversarial examples with original voices as reference

Figure 13: SPL distribution of adversarial perturbation and MCDs of adversarial examples.

and larger sensing range, thus introducing more ambient noises. Moreover, compared with *NoAug*, *PhyTalker* achieves 44.4% higher ASRs on average. This result demonstrates that the channel augmentation does improve the robustness of adversarial perturbations in physical attacks. As for *SingleAug*, its ASR is 97.8% on the seen device model (i.e., CLIP3-M10W), but rapidly decreases to 53.9%, 51.6% and 42.5% under other unknown device models, respectively. This indicates introducing sufficient channel responses for augmentation can significantly improve the robustness of *PhyTalker*.

Figure 11(b) shows ASRs in different environments (i.e., Lab, Office, Study and Cafe). It can be observed that ASRs of *PhyTalker* are 85.5%,88.2%, 83.2% and 78.3%, higher than *NoAug* by 19.0%, 20.8%, 23.9% and 37.1%, respectively, indicating that *PhyTalker* is robust to different environments. But for *SingleAug*, its ASRs are 92.4%, 80.6%, 78.2% and 61.1% under the four environments, exhibiting less variances. This result indicates environments induce slighter interference than device models, which is consistent with our previous observation as shown in Figure 4(b). Especially, the noisy environment, Cafe scenario, greatly affects the attack effect under *NoAug* setup. But our proposed channel augmentation algorithm effectively resists it and maintains the attack's effectiveness.

4.5 Performance of Cross-model Transferability

We further evaluate the transferability of *PhyTalker* across different SR systems. In addition to the two available cross types among the models in Table 2 (cross training dataset and model architecture), we retrain model A using 13*3 dynamic MFCCs with 50ms duration and 10ms hop as input feature, which contains first-order and second-order difference coefficients extracted from a larger frame thus reflecting more abundant dynamic information of the speech. Figure 12(a) shows ASRs of *PhyTalker* with ensemble and single models under five different cross types (i.e., white-box, cross training data, cross input feature, cross model architecture and cross all the three factors). We can see that ASRs with ensemble model are 94.8%, 91.5% 89.6%, and 75.4%, which are higher than those of the attack with the single model by 28.0%, 27.0%, 37.5%, and 36.9%, respectively. This indicates that the model ensemble contributes to improving the robustness of attacking different black-box models than a single model. Moreover, it can be also observed that the attack cross all three factors exhibit worst ASR than other cross types. But even under such a situation that is closest to real physical attacks, *PhyTalker* can achieve the ASR of 75.4%, validating its effectiveness.

Figure 12(b) shows ASRs of *PhyTalker* across both of model and data under ensemble method with different number of models. We can see that the average ASRs are 48.3%, 69.5%, 83.5% and 85.5% under 1, 2, 3 and 4 models for ensemble, respectively, exhibiting an increasing trend with the increase of model numbers. But the result also shows that the ASR increases gradually becomes little as the model number increases. This is because the ensemble model can only construct a more generalized solution space, instead of the exact solution space of the target model. But even under such an incomplete space, *PhyTalker* can also achieves 85.5% ASR on average under 4 models in ensemble, demonstrating its robustness under black-box models.

4.6 Performance of Human Imperceptibility

We also evaluate the human imperceptibility of *PhyTalker* with both objective experiments and subjective human study.

We first measure the SPL distribution round the attack device when broadcasting perturbations generated by *PhyTalker*, for that SPL represents the strength of voice, and it intuitively reflects the imperceptibility of the perturbation without human voice. The SPLs are measured by a SMART SENSER AR844 (30~80 dB, A-weight). Figure 13(a) shows SPL distribution around the attack device under different distances and angles. We can see that the SPLs in front of the attack device (i.e., around $0^\circ \sim 30^\circ$) are significantly higher than that at other angles. When the angle is greater than 30°, the SPL beyond 1m is lower than 39.2dB, only 0.5dB higher than that in ambient environments. This result indicates a surrounding person can perceive the perturbation only if he/she appears in specific angles to the attack device. Moreover, in the perspective of distance, the maximum SPL of 45.1 dBA appears at the distance of around 0.5m. But when the distance increases to 2m, the SPL rapidly attenuates to 38.9dB. Considering the common social distance by WHO (i.e., 1m) [57], such a small SPL is difficult to perceive by surrounding people. Also, the adversary can intentionally control his/her device's direction to avoid the attention of surroundings.

We further evaluate the human-audibility of the adversarial examples in physical domain with MCDs. In the experiment, we introduce additional original voices propagating over the air (i.e., *Origin*) and *FakeBob* as baselines. Then, we derive MCDs for *PhyTalker* and two other baselines with the same original voices as the reference. Figure 13(b) shows MCDs of *PhyTalker*, *Origin* and *FakeBob* respectively. We can see that the average MCDs are 2.45dB, 2.24dB and

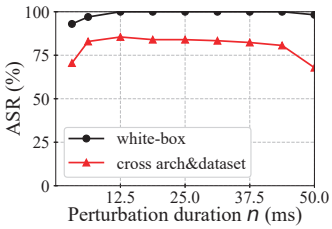


Figure 14: ASRs under different perturbation durations n (ϵ=0.02, d=5)

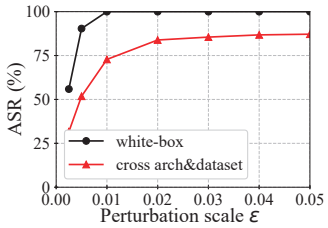


Figure 15: ASRs under different perturbation scales ϵ (n=12.5, d=5)

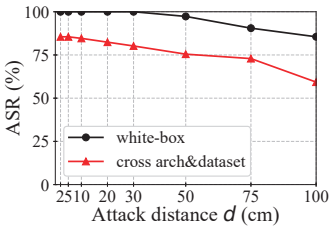


Figure 16: ASRs under different attack distances d (n=12.5, ϵ=0.02)

4.15dB for the three methods respectively. Compared with *Origin*, MCD of *PhyTalker* only increase 0.21dB, indicating that the distortion introduced by *PhyTalker* is similar to that of environmental noises. On the other hand, the average MCD of *PhyTalker* is 1.7dB lower than that of *FakeBob*. These results further demonstrate the imperceptibility of *PhyTalker*, which is also consistent with the analysis in Section 4.2.

We also conduct the subjective human study on the imperceptibility of *PhyTalker*. In the experiment, we recruit 10 volunteers including 5 males and 5 females aging from 22 to 28. They are asked to listen to 50 pairs of normal voice samples and corresponding adversarial examples. All the audio pairs are recorded at 1m away from the experimental platform. We then require the volunteers to report their intuitive sense of the audibility similarity between normal samples and adversarial examples with a score ranging 1~5 (a higher score indicates better audibility with less noisy perceptibility). Note that the experiments on volunteers are validated by the Institutional Review Board (IRB) in our university. The result shows that the average score of *PhyTalker* is 4.6, indicating the imperceptibility of adversarial examples for human audibility.

4.7 Ablation Study

We also investigate the impact of several key variables on the performance of *PhyTalker*, including the perturbation duration, perturbation scale, and attack distance. In the experiments, we select the model D with average overall performance as the target SR system for simplicity and explore the easiest white-box setting and the most difficult cross architecture&dataset black-box setting, i.e., ensemble model B, C, H, I as substitute system for the attack.

Perturbation duration. Figure 14 shows ASRs of *PhyTalker* with different durations under the white-box and black-box settings. We can see that the ASR is less affected by the perturbation duration under the white-box setting, while there are about 10% ASR drops with too short or too long perturbations under the black-box setting. This result indicates the trade-off between the perturbation coverage and segmentation, i.e., longer perturbations carry more adversarial information but may be truncated by the frame segmentation during signal processing, while shorter perturbations do the opposite. Hence, we select 12.5ms as perturbation duration in the implementation of *PhyTalker*.

Perturbation scale. Figure 15 shows ASRs of *PhyTalker* with different perturbation scales under the white-box and black-box settings. As the perturbation scale increases, the ASRs under both settings exhibit a rapid growth, then go stable after a turning point (i.e., ϵ=0.01 and ϵ=0.02), and finally reach 100% and 85.5%, respectively. This result indicates a better attack performance under a larger perturbation scale. However, a larger perturbation scale would also produce severer audibility distortion. Considering the limited performance improvement after the turning point and the acoustic signal attenuation at the attack distance, we set 0.02 as the perturbation scale in the implementation of *PhyTalker* to balance the attack performance and imperceptibility.

Attack distance. Considering that the actual impersonation is induced by the broadcasting perturbation, this experiment investigates the impact of the loudspeaker-SR distance on *PhyTalker*. Figure 16 shows ASRs of *PhyTalker* with different distances between the loudspeaker and SR front-end under the white-box and black-box settings. Note that we do not re-train or calibrate the model according to to adapt to different distances, and maintain the volume to an equally low level (i.e., 38.9dBA SPL at 1 m distance) for imperceptibility with the growth of the attack distance. Figure 16 shows that, as the attack distance increases, ASRs of both attack settings drop gradually, due to a longer distance involving severe energy attenuation and multi-path interference from environmental factors such as sound propagation delay. However, even at the worst case of 100cm distance, the ASRs still achieve 85.5% and 76.3% under the white-box and black-box settings, respectively.

4.8 In-the-wild Experiment

In addition to the simulation experiment using a loudspeaker, we further conduct an in-the-wild experiment involving human speakers for the evaluation. Figure 17 shows the experimental setup of the in-the-wild experiment. We recruit 10 volunteers including six males (i.e., P0~P5) and four females (i.e., P6~P9) to act as the adversary respectively. Each volunteer sits 50cm away from the ASR front-end, and carries an attack device connected with a loudspeaker and a lavalier microphone. The loudspeaker is placed 5cm away from ASR front-end to broadcast perturbations. The microphone is attached to the volunteer near his/her mouth to record the live-streaming voices. Moreover, as shown in Figure 10, we find different synchronization mechanisms probably affect the attack performance. Hence, in this experiment, except for the previous mentioned statics RPS synchronization mechanism, we further introduce another *pre-record RPS*, i.e., the RPS with phoneme duration is extracted from a pre-recorded speech, which is obtained by the adversary in advance. Hence, in each experiment, each volunteer is first required to say an arbitrary 15-minute text for subphoneme-level perturbation training, then to say ten speech commands five times, one for extracting pre-record RPS, and four for different test settings (i.e., attacking a male/female user with original/pre-record

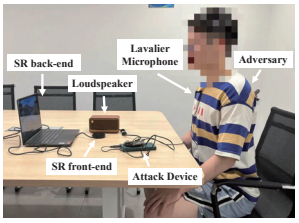


Figure 17: Experimental setup of in-the-wild experiment.

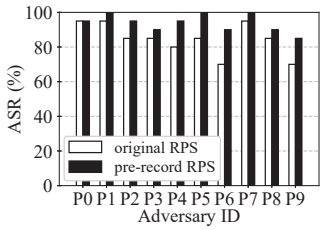


Figure 18: ASRs of ten volunteers with two different synchronization mechanisms.

RPS). Figure 18 shows the ASRs of the ten volunteers with different synchronization mechanisms. We can see that *PhyTalker* with the *original RPS* achieves 84.5% ASR on average, with a standard deviation of 8.8%. Compared with the performance of variable-controlled experiments in Table 3, the in-the-wild result slightly declines them by 1.0% ASR. In addition, it can be observed that *PhyTalker* with the *pre-record RPS* is more effective, achieving an ASR improvement of 9.5% than that of *original RPS*. These results encourage the adversary to employ *pre-record RPS* in real physical attacks.

To further validate the performance of *PhyTalker* on real commercial systems, we launch *PhyTalker* attacking on Apple Siri[3] of an iPhone 13. In this experiment, we recruit five volunteers to enroll in the system, and another one to issue the attack. Before the experiment, we require the adversary to try to trigger Siri without the adversarial perturbation, whose results show that the adversary cannot activate Siri, validating that Siri realizes the speaker recognition. Since Siri is a text-dependent SR system, we select five common commands for the attack, i.e., "Hey Siri, play some music", "Hey Siri, what's the weather like today", "Hey Siri, send a message", "Hey Siri, open the light" and "Hey Siri, open the TV". The experimental results show that *PhyTalker* successfully enables the adversary to activate Siri enrolled by the five victims under all the five commands, which requires 1.6 attack attempts on average. This further supports the effectiveness of *PhyTalker* on commercial systems in physical domain.

5 DISCUSSIONS

Interference by Various Voice Speed. Humans' voice speed varies depending on different factors, such as speech texts, speakers' physiological and emotional state, surrounding environments. Many factors introduce significant speed changes, probably reducing the accuracy of the phoneme duration estimation, thus resulting in unsatisfactory real-time adversarial perturbation injection. A representative example includes a speaker is scared by an unpredictable event and thus suddenly increases his/her voice speed. However, considering the impersonation attack is usually intentional, the adversary could control his/her speed at a stable level during an attack, rather than changing frequently. Hence, *PhyTalker* could employ EWMA algorithm to adjust the perturbation for adapting to the adversary's voice speed.

Performance Degradation under Unusual Phonemes. The occurrence frequency of each phoneme in different conversations is distinct from each other. For example, in LibriSpeech, the average number of [ʌ] and [ɔɪ] in an utterance are 13.10 and 0.12, respectively. Such a difference indicates different amounts of corpus for subphoneme-level perturbation optimization. Especially for the unusual phonemes (i.e., phonemes with low occurrence frequency), the fewer samples probably lead to an inadequate perturbation optimization, resulting in poorer attack performance. But the adversary can select a particular corpus for perturbation optimization based on his/her intended query to the SR. The particular corpus is designated to involve more phonemes that would occur in the physical attack, so as to ensure the attack performance.

Countermeasures. Based on the characteristics of *PhyTalker*, we provide several potential countermeasures in terms of three stages of SR systems, i.e., pre-processing, adversarial learning, and post-detection.

(1) *Pre-processing* is a series of straightforward methods to mitigate adversarial perturbation before feeding into SR models, including low-pass filtering, downsampling, re-quantization, and compression, etc. Among them, recent work [45] reports low-pass filtering is the most processing method for defending against audio adversarial example attacks, which also applies to our attacks theoretically. The voice frequency band concentrates on [300, 3,400]Hz approximately [43], but the perturbation's spectrum distributes in a higher frequency band, mainly from 2,000Hz to 8,000Hz. Hence, a pre-processing method that destroys high-frequency information, such as downsampling and low-pass, can be applied to received audio to effectively mitigate the adversarial perturbation.

(2) *Adversarial learning* is also an effective countermeasure as validated in SOTA works [14, 23], which introduces different kinds of negative samples in the model training phase to enhance its generalization capability in resisting adversarial examples in the model processing stage. Specifically, during the training phase of SR models, the SR service provider could pre-generate various adversarial examples by implementing existing generation approaches, so that the vulnerabilities in the decision boundary that are easy to exploited by adversarial example attacks, could be fixed. To this end, our proposed *PhyTalker* actually provides a new kind of adversarial example generation approach, which serves as the basis of negative sample generation for adversarial training, thus enabling SR models to effectively mitigate such adversarial examples.

(3) *Post-detection* turns to monitor the existence of crafted perturbations in received signals, serving as a reference for the legitimacy of SR system output in the decision-making stage, instead of directly mitigating them. Spectrum detection is a representative one to resist *PhyTalker*, which performs Short-Time Fourier Transform (STFT) on the signals after high-pass filtering, then counts the number of same energy peaks in adjacent frames in the audio, and finally determines whether it is an adversarial sample by comparing with a threshold. Due to the repeated broadcasting manner of subphoneme-level perturbations, significant repeated patterns would appear in the spectrum, especially a distinct spectrum pattern from normal voices in the high-frequency region. By detecting such a difference posterior to the recognition, SR systems are able to decline the spoofing request from *PhyTalker*.

6 RELATED WORK

Similar to visual adversarial example attacks, researches on audio adversarial examples are first explored in digital domain, where generated examples are directly injected into the target system without the consideration of air-borne propagation. Early works [7, 18, 26, 54, 55, 60] explore the feasibility of launching white-box attacks, i.e., the target system is transparent to the adversary. They leverage classical gradient-based methods (e.g., FGSM [16], BIM [28], PGD [38]) to optimize adversarial examples. Based on the composability and stability of phoneme, Turner et al. [52] propose to map and replace the phonemes in the original speaker's speech with those of the target speaker. Recent work PhoneyTalker [9] employs a generative model to generate universal adversarial perturbations for each phoneme automatically. However, these attacks still show little threat to real systems due to their impractical assumption of digital and white-box settings.

To realize a practical audio adversarial example attack in physical domain, the following works make efforts to investigate over-the-air scenario, black-box setting and live-streaming context, respectively. Table 4 shows the tasks and characteristics of these works.

The most significant difference between attacks in digital and physical domains is whether the generated examples are injected over the air. When signals propagate through physical channels, audio adversarial examples suffer from various practical effects (e.g., device non-linearity, multi-path effect, and noise interference). To alleviate the interference of channel, data augmentation[39] and domain adaptation[21] are proposed in the domain of speech recognition. To enable adversarial examples in over-the-air scenarios, recent studies [10, 34, 35, 58, 59, 61] exploit channel simulation techniques (e.g., Room Impulse Response [50]) to compensate the signal distortion during the air-borne propagation. However, these works ignore the channel distortion caused by devices instead of the environments, which has been demonstrated to significantly downgrade the attack robustness in our work.

Moreover, in a physical attack, the adversary usually has no prior knowledge about the target system. Some works [8, 11, 24, 51] adopt a query-based scheme to estimate gradient for example generation. Especially, FakeBob [8] introduces a Natural Evolution Strategy (NES [20]) to seek an effective gradient direction with the queried decision result, which is proven to be efficient to compromises unknown commercial systems. However, numerous queries are required for accurate gradient estimation, which easily arouses the target's awareness. Other works [33, 42, 62] turn to exploit the transferability [37] of adversarial examples. They construct a local substitute model to simulate the behaviors of the unknown target model for example generation. However, such transfer-based attacks exhibit weak generalization on attacking dissimilar models.

To avoid target awareness, live-streaming attack, instead of replaying recorded voices, exhibits stronger practicality in physical attacks. Recent studies [32, 59, 61] propose to repeat universal adversarial perturbations with fixed length for any speech content. However, due to excessive optimization objectives and the regardless of specific speech content, it's difficult to train a universal adversarial perturbation with strong attack ability in the original limited feature space that allowed to be modified. Hence, a more relaxed feature space, i.e., a larger perturbation scale, is in need, thus sacrificing the audibility (e.g., AdvPulse only achieves 8.7dB SNR in white-box and digital experiments). More recently, Chiquier et al. [12] propose a predictive model with real-time constraints to generate forward-looking perturbations for future signal, thus realizing a live-streaming synchronization between the voice and corresponding perturbation, but its specific real-time performance on mobile devices under real-human setting is still unclear.

Different from these existing studies focusing on digital or partially featured physical attacks, our work comprehensively summarizes the practical issues, including over-the-air scenario, black-box setting and live-streaming context, in a physical attack. Especially, as discussed in Section 4.2, our attack outperforms the recent SOTA works *FakeBob*[8] and *AdvPulse*[35] on efficiency and imperceptibility, respectively. By constructing subphoneme-level adversarial perturbations with the multi-model ensemble, cross-channel augmentation and live-streaming synchronization, we realize a more practical audio adversarial example attack.

Table 4: Tasks and characteristics of related works.

Research work	Task	Live-streaming	Over-the-air	Black-box
Zhang et al.[62]	speech	✗	✗	✓
PhoneyTalker [9]	speaker	✗	✗	✓
Metamorph[10]	speech	✗	✓	✗
Imperio[47]	speech	✗	✓	✗
Li et al.[34]	speaker	✗	✓	✗
Xie et al.[58]	both	✗	✓	✗
Devil's whisper[11]	speech	✗	✓	✓
FakeBob[8]	speaker	✗	✓	✓
AdvPulse[35]	both	✓	✓	✗
Xie et al.[59]	speaker	✓	✓	✗
Zhang et al.[61]	speaker	✓	✓	✗
Chiquier et al.[12]	speech	✓	✗	✗
PhyTalker	speaker	✓	✓	✓

7 CONCLUSION

In this paper, we propose *PhyTalker*, a live-streaming, cross-channel and black-box adversarial example attack on speaker recognition in physical domain. *PhyTalker* can be divided into the offline perturbation optimization and online attacking phases. In the offline phase, *PhyTalker* generates the subphoneme-level perturbations for different phonemes as a dictionary, where the channel impulse response and model ensemble method are introduced to improve its channel robustness and transferability. After that, in the online phase, *PhyTalker* estimates the type and duration of the latest recorded phoneme, and prepares the corresponding perturbations. And then the selected perturbations are broadcast for the injection on the adversary's live-streaming voices in physical domain. Extensive experiments under large-scale corpus in real scenarios validate that *PhyTalker* can successfully spoof mainstream SR systems while remaining inaudible to surrounding people in physical domain.

ACKNOWLEDGMENTS

This research is sponsored by National Key R&D Program of China (2020AAA0107700), National Natural Science Foundation of China (62102354, 62032021, 62122066, 62172359, 61972348, 62172277), Fundamental Research Funds for the Central Universities (2021FZZX001-27).

REFERENCES

[1] FAKEBOB adversarial attack, Tom Dorr, Golfer Chen, and Pengfei Gao. 2019. FAKEBOB. https://github.com/FAKEBOB-adversarial-attack/FAKEBOB.

[2] Amazon Help & Customer Service. 2022. What Is Alexa Voice ID? https://www.amazon.com/gp/help/customer/display.html?nodeId=202199440.

[3] Apple. 2022. Apple Siri. https://www.apple.com/sg/siri/.

[4] Mathieu Bernard and Hadrien Titeux. 2021. Phonemizer: Text to Phones Transcription for Multiple Languages in Python. Journal of Open Source Software 6, 68 (2021), 3958. https://doi.org/10.21105/joss.03958

[5] Raghav Bharadwaj. 2019. Voice and Speech Recognition in Banking – What's Possible Today. https://emerj.com/ai-sector-overviews/voice-speech-recognition-banking/.

[6] Frédéric Bimbot, Jean-François Bonastre, Corinne Fredouille, Guillaume Gravier, Ivan Magrin-Chagnolleau, Sylvain Meignier, Téva Merlin, Javier Ortega-Garcia, Dijana Petrovska-Delacrétaz, and Douglas A. Reynolds. 2004. A Tutorial on Text-Independent Speaker Verification. EURASIP J. Adv. Signal Process. 2004, 4 (2004), 430–451.

[7] Nicholas Carlini and David A. Wagner. 2018. Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. In Proceedings of SP Workshops. IEEE Computer Society, San Francisco, CA, USA, 1–7.

[8] Guangke Chen, Sen Chen, Lingling Fan, Xiaoning Du, Zhe Zhao, Fu Song, and Yang Liu. 2021. Who is Real Bob? Adversarial Attacks on Speaker Recognition Systems. In Proceedings of SP. IEEE, Los Alamitos, CA, USA, 55–72.

[9] Meng Chen, Li Lu, Zhongjie Ba, and Kui Ren. 2022. PhoneyTalker: An Out-of-the-Box Toolkit for Adversarial Example Attack on Speaker Recognition. In Proceedings of INFOCOM. IEEE, Virtual Event, 1419–1428.

[10] Tao Chen, Longfei Shangguan, Zhenjiang Li, and Kyle Jamieson. 2020. Metamorph: Injecting Inaudible Commands into Over-the-air Voice Controlled Systems. In Proceedings of NDSS. The Internet Society, San Diego, California, USA.

[11] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. 2020. Devil's Whisper: A General Approach for Physical Adversarial Attacks against Commercial Black-box Speech Recognition Devices. In Proceedings of USENIX Security Symposium. USENIX Association, 2667–2684.

[12] Mia Chiquier, Chengzhi Mao, and Carl Vondrick. 2022. Real-Time Neural Voice Camouflage. In Proceedings of ICLR. OpenReview.net, Virtual Event.

[13] F. A. Rezaur Rahman Chowdhury, Quan Wang, Ignacio Lopez-Moreno, and Li Wan. 2018. Attention-Based Models for Text-Dependent Speaker Verification. In Proceedings of ICASSP. IEEE, Calgary, AB, Canada, 5359–5363.

[14] Mohammad Esmaeilpour, Patrick Cardinal, and Alessandro Lameiras Koerich. 2021. Class-Conditional Defense GAN Against End-To-End Speech Attacks. In Proceedings of ICASSP. IEEE, Toronto, ON, Canada, 2565–2569.

[15] Chao Gao, Guruprasad Saikumar, Amit Srivastava, and Premkumar Natarajan. 2011. Open-set speaker identification in broadcast news. In Proceedings of ICASSP. IEEE, Prague, Czech Republic, 5280–5283.

[16] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In Proceedings of ICLR. OpenReview.net, San Diego, CA, USA.

[17] Google Assistant Help. 2022. Teach Google Assistant to recognize your voice with Voice Match. https://support.google.com/assistant/answer/9071681.

[18] Keita Goto and Nakamasa Inoue. 2020. Quasi-Newton Adversarial Attacks on Speaker Verification Systems. In Proceedings of APSIPA ASC. IEEE, Auckland, New Zealand, 527–531.

[19] Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Networks 18, 5-6 (2005), 602–610.

[20] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. 2018. Black-box Adversarial Attacks with Limited Queries and Information. In Proceedings of ICML, Vol. 80. IEEE, Stockholmsmässan, Stockholm, Sweden, 2142–2151.

[21] Md Tamzeed Islam and Shahriar Nirjon. 2021. Sound-Adapter: Multi-Source Domain Adaptation for Acoustic Classification Through Domain Discovery. In Proceedings of IPSN. ACM, Nashville, TN, USA, 176–190.

[22] ISO. 2009. Measurement of room acoustic parameters-part 1: Performance spaces. Standard. International Organization for Standardization.

[23] Arindam Jati, Chin-Cheng Hsu, Monisankha Pal, Raghuveer Peri, Wael AbdAlmageed, and Shrikanth Narayanan. 2021. Adversarial attack and defense strategies for deep speaker recognition systems. Comput. Speech Lang. 68 (2021), 101199.

[24] Shreya Khare, Rahul Aralikatte, and Senthil Mani. 2019. Adversarial Black-Box Attacks on Automatic Speech Recognition Systems Using Multi-Objective Evolutionary Optimization. In Proceedings of Interspeech. ISCA, Graz, Austria, 3208–3212.

[25] Aldebaro Klautau. 2001. ARPABET and the TIMIT alphabet. (2001).

[26] Felix Kreuk, Yossi Adi, Moustapha Cissé, and Joseph Keshet. 2018. Fooling End-To-End Speaker Verification With Adversarial Examples. In Proceedings of ICASSP. IEEE, Calgary, AB, Canada, 1962–1966.

[27] R. Kubichek. 1993. Mel-cepstral distance measure for objective speech quality assessment. In Proceedings of PACRIM, Vol. 1. IEEE, 125–128.

[28] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2017. Adversarial examples in the physical world. In Proceedings of ICLR. OpenReview.net, Toulon, France.

[29] Anthony Larcher, Kong-Aik Lee, Bin Ma, and Haizhou Li. 2014. Text-dependent speaker verification: Classifiers, databases and RSR2015. Speech Commun. 60 (2014), 56–77.

[30] Vladimir I. Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions and reversals. Dokl. Akad. Nauk SSSR (1966).

[31] Chao Li, Xiaokong Ma, Bing Jiang, Xiangang Li, Xuewei Zhang, Xiao Liu, Ying Cao, Ajay Kannan, and Zhenyao Zhu. 2017. Deep Speaker: an End-to-End Neural Speaker Embedding System. CoRR abs/1705.02304 (2017).

[32] Jiguo Li, Xinfeng Zhang, Chuanmin Jia, Jizheng Xu, Li Zhang, Yue Wang, Siwei Ma, and Wen Gao. 2020. Universal Adversarial Perturbations Generative Network For Speaker Recognition. In Proceedings of ICME. IEEE, London, UK, 1–6.

[33] Xu Li, Jinghua Zhong, Xixin Wu, Jianwei Yu, Xunying Liu, and Helen Meng. 2020. Adversarial Attacks on GMM I-Vector Based Speaker Verification Systems. In Proceedings of ICASSP. IEEE, Barcelona, Spain, 6579–6583.

[34] Zhuohang Li, Cong Shi, Yi Xie, Jian Liu, Bo Yuan, and Yingying Chen. 2020. Practical Adversarial Attacks Against Speaker Recognition Systems. In Proceedings of HotMobile. ACM, Austin, TX, USA, 9–14.

[35] 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 CCS. ACM, Virtual Event, USA, 1121–1134.

[36] Tingting Liu and Shengxiao Guan. 2014. Factor analysis method for text-independent speaker identification. Journal of Software 9, 11 (2014), 2851–2860.

[37] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2017. Delving into Transferable Adversarial Examples and Black-box Attacks. In Proceedings of ICLR. OpenReview.net, Toulon, France.

[38] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In Proceedings of ICLR. OpenReview.net, Vancouver, BC, Canada.

[39] Akhil Mathur, Tianlin Zhang, Sourav Bhattacharya, Petar Velickovic, Leonid Joffe, Nicholas D. Lane, Fahim Kawsar, and Pietro Liò. 2018. Using deep data augmentation training to address software and hardware heterogeneities in wearable and smartphone sensing devices. In Proceedings of IPSN, Luca Mottola, Jie Gao, and Pei Zhang (Eds.). IEEE / ACM, Porto, Portugal, 200–211.

[40] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. 2016. TUT database for acoustic scene classification and sound event detection. In Proceedings of EUSIPCO. IEEE, Budapest, Hungary, 1128–1132.

[41] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. 2017. VoxCeleb: A Large-Scale Speaker Identification Dataset. In Processings of Interspeech, Francisco Lacerda (Ed.). ISCA, Stockholm, Sweden, 2616–2620.

[42] Paarth Neekhara, Shehzeen Hussain, Prakhar Pandey, Shlomo Dubnov, Julian J. McAuley, and Farinaz Koushanfar. 2019. Universal Adversarial Perturbations for Speech Recognition Systems. In Proceedings of Interspeech. ISCA, Graz, Austria, 481–485.

[43] Institute of Telecommunication Sciences. 1996. voice frequency. https:/www.its.bldrdoc.gov/fs-1037/dir-039/_5829.htm.

[44] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In Processings of ICASSP. IEEE, South Brisbane, Queensland, Australia, 5206–5210.

[45] Krishan Rajaratnam, Kunal Shah, and Jugal Kalita. 2018. Isolated and Ensemble Audio Preprocessing Methods for Detecting Adversarial Examples against Automatic Speech Recognition. In Proceedings of ROCLING. Hsinchu, Taiwan, 16–30.

[46] Douglas D. Rife and John Vanderkooy. 1989. Transfer-function measurement with maximum-length sequences. Journal of the Audio Engineering Society 37, 6 (june 1989), 419–444.

[47] Lea Schönherr, Thorsten Eisenhofer, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. 2020. Imperio: Robust Over-the-Air Adversarial Examples for Automatic Speech Recognition Systems. In Proceedings of ACSAC. ACM, Austin, TX, USA, 843–855.

[48] Seeed. 2018. ReSpeaker Core v2.0. https://wiki.seeedstudio.com/ReSpeaker_Core_v2.0/.

[49] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. 2018. X-Vectors: Robust DNN Embeddings for Speaker Recognition. In Proceedings of ICASSP. IEEE, Calgary, AB, Canada, 5329–5333.

[50] Guy-Bart Stan, Jean-Jacques Embrechts, and Dominique Archambeau. 2002. Comparison of different impulse response measurement techniques. Journal of the Audio Engineering Society 50, 4 (2002), 249–262.

[51] Rohan Taori, Amog Kamsetty, Brenton Chu, and Nikita Vemuri. 2019. Targeted Adversarial Examples for Black Box Audio Systems. In Proceedings of SP Workshops. IEEE, San Francisco, CA, USA, 15–20.

[52] Henry Turner, Giulio Lovisotto, and Ivan Martinovic. 2019. Attacking Speaker Recognition Systems with Phoneme Morphing. In Proceedings of ESORICS, Kazue Sako, Steve A. Schneider, and Peter Y. A. Ryan (Eds.), Vol. 11735. Springer,

471–492.
[53] Ehsan Variani, Xin Lei, Erik McDermott, Ignacio Lopez-Moreno, and Javier Gonzalez-Dominguez. 2014. Deep neural networks for small footprint text-dependent speaker verification. In Proceedings of ICASSP. IEEE, Florence, Italy, 4052–4056.
[54] Jesús Villalba, Yuekai Zhang, and Najim Dehak. 2020. x-Vectors Meet Adversarial Attacks: Benchmarking Adversarial Robustness in Speaker VeEication. In Proceedings of Interspeech. ISCA, Shanghai, China, 4233–4237.
[55] Qing Wang, Pengcheng Guo, and Lei Xie. 2020. Inaudible Adversarial Perturbations for Targeted Attack in Speaker Recognition. In Proceedings of Interspeech. ISCA, Shanghai, China, 4228–4232.
[56] WeChat. 2015. Voiceprint: The New WeChat Password. https://blog.wechat.com/2015/05/21/voiceprint-the-new-wechat-password/.
[57] WHO. 2019. Advice for the public: Coronavirus disease (COVID-19). https://www.who.int/emergencies/diseases/novel-coronavirus-2019/advice-for-public.
[58] Yi Xie, Zhuohang Li, Cong Shi, Jian Liu, Yingying Chen, and Bo Yuan. 2021. Enabling Fast and Universal Audio Adversarial Attack Using Generative Model. In Proceedings of AAAI. AAAI Press, Virtual Event, 14129–14137.
[59] Yi Xie, Cong Shi, Zhuohang Li, Jian Liu, Yingying Chen, and Bo Yuan. 2020. Real-Time, Universal, and Robust Adversarial Attacks Against Speaker Recognition Systems. In Proceedings of ICASSP. IEEE, Barcelona, Spain, 1738–1742.
[60] 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 Proceedings of USENIX Security Symposium. USENIX Association, Baltimore, MD, USA, 49–64.
[61] Weiyi Zhang, Shuning Zhao, Le Liu, Jianmin Li, Xingliang Cheng, Thomas Fang Zheng, and Xiaolin Hu. 2021. Attack on Practical Speaker Verification System Using Universal Adversarial Perturbations. In Proceedings of ICASSP. IEEE, Toronto, ON, Canada, 2575–2579.
[62] Yuekai Zhang, Ziyan Jiang, Jesús Villalba, and Najim Dehak. 2020. Black-Box Attacks on Spoofing Countermeasures Using Transferability of Adversarial Examples. In Proceedings of Interspeech. ISCA, Shanghai, China, 4238–4242.