Trustworthy User-Machine Interactions

by

Shimaa Ahmed

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Electrical and Computer Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2024

Date of final oral examination: 01/10/2024

The dissertation is approved by the following members of the Final Oral Committee:

Kassem Fawaz, Associate Professor, Electrical and Computer Engineering
Parmesh Ramanathan, Professor, Electrical and Computer Engineering
Suman Banerjee, Professor, Computer Sciences
Somesh Jha, Professor, Computer Sciences

To my family and friends for their unwavering support and encouragement.

Acknowledgments

الحمد لله رب العالمين

All praise is for Allah, the Lord of all worlds.

— The Qur'an, 1:2

First and foremost, I extend my deepest gratitude to Allah for giving me strength, patience, and serenity throughout my academic journey. I am grateful for all the blessings that have guided me and contributed to the successful completion of this work.

I would like to express my gratitude and appreciation to my advisor, Kassem Fawaz, for his genuine support, invaluable mentorship, and unwavering guidance. I am eternally grateful for the opportunities he provided, especially during the first two years of my Ph.D., when the odds were against me. He introduced me to this eloquent Arabic saying: "إن هبت أمراً فقع فيه", which means "When you fear a matter, dive straight into it." It resonated perfectly with me. I am profoundly thankful for his encouragement to push my boundaries and explore my full potential. Kassem has not only helped me become an independent researcher but has also fostered my development both technically and mentally. He has granted me access to an amazing professional network and has consistently provided the necessary resources to carry out my work.

To my Ph.D. committee, Prof. Parmesh Ramanathan, Prof. Suman Banerjee, and Prof. Somesh Jha, I extend my sincere thanks for their valuable feedback, encouragement, and rigorous assessment of my work. Their diverse perspectives and constructive critiques have significantly contributed to the refinement of my research and scholarly development. I would also like to thank Prof. Nicolas Papernot. He was an amazing advisor and a very friendly mentor whose insightful comments and discussions shaped the way I think about research in many aspects.

A special word of thanks goes to my labmates and collaborators, whose support, care, and shared experiences have made the Ph.D. journey more enjoyable. The stimulating discussions, constructive critique, and collective brainstorming meetings have greatly enriched my PhD work. I am grateful for the learning experience and the lasting friendships formed in the process. I would like to single out my PhD mentors: Amrita Roy Chowdhury, Ilia Shumailov, and Varun Chandrasekaran. Their guidance and encouragement were instrumental in my growth, I still reach out to them for advice in many of my professional decisions. I would also like to extend a special thanks to my first mentee, Yash Wani, for his insightful perspectives, commitment, and unique skills that were very valuable to our research.

I am also indebted to my friends in Madison: Mariam, Hassnaa, Nada, and Nour whose unwavering support, encouragement, and companionship have been a source of comfort and joy. Their presence and genuine friendship made my time in Madison memorable and helped me keep my sanity, especially during the COVID time. A heartfelt thank you to my best friend, Sally, whose friendship has been a constant source of strength and happiness. Sally is my personal advisor, she helped me navigate the challenging times, celebrate the achievements, and enjoy the normal days. Sally, your unwavering support and encouragement, your sarcasm and humor, and your rational judgment have been invaluable to me.

Last, but certainly not least, my deepest gratitude goes to my family. To my mother and brothers, to my late father and grandmother Fatma, your love, sacrifice, and unwavering belief in me have been the foundation of my strength and perseverance. Your support has been my greatest motivation. To my extended family, thank you for your love, endless encouragement, and for always believing in me. Your support has been a source of great comfort and strength.

This journey would not have been possible without each and every one of you. Thank you from the bottom of my heart.

Contents

List of Tables

List of Figures

Abstract

In an era where smart devices and Machine Learning as a Service (MLaaS) are ubiquitous, and data is abundantly available, technologies such as speech recognition, speaker identification, and face recognition have become integral to user-machine interactions. These technologies, developed primarily for performance enhancement and user experience improvement, are pivotal in how humans interact with machines. However, their widespread adoption has brought forth significant privacy, security, and integrity challenges. These include unauthorized access to private data stored in the cloud, accidental activations of smart devices, unsolicited biometric data collection, and susceptibility to impersonation attacks. Furthermore, these technologies demonstrate failure modes and spurious correlations that disproportionately affect certain demographic groups.

This thesis delves into a multi-faceted approach aimed at reinforcing more secure and reliable user-machine interactions. Our objectives are threefold. First, we analyze the emerging privacy and security threats associated with machine learning technologies, especially those involving biometric data. Second, we scrutinize and challenge the current standards of security in biometric authentication, revealing vulnerabilities that have been largely overlooked. Finally, we develop and implement practical solutions designed to mitigate these risks, thereby maintaining the utility and convenience of these technologies. Through this thesis, we aspire to establish a new standard in the development and deployment of user-machine interaction technologies, ensuring they are not only efficient and convenient but also secure and equitable for all users.

Chapter 1

Introduction

Machine Learning (ML) systems power our everyday interactions with digital services and personal devices. Social media platforms, smart devices, governments, and businesses employ ML for a myriad of tasks, including identity recognition and authentication, speech transcription, personalization of user experiences, and targeted advertising. More recently, emerging generative AI technologies have revolutionized how we create and interact with digital content, offering unprecedented capabilities in generating realistic media including images, text, and speech. The widespread adoption of ML systems, however, comes at a considerable societal cost in terms of privacy, fairness, and trust. Generative AI also brings new challenges in areas like authenticity verification and ethical use.

ML systems consume massive amounts of data from their users for both main task inference and the ongoing training and benchmarking of the underlying models. This raises concerns about data privacy, as individuals' information is collected, analyzed, and potentially shared without their consent. Often, the data is intrusive, encompassing sensitive details about users' identities, behaviors, and actions. Additionally, ML models can deduce more information about users than what is intended for their primary function. Consequently, this situation forces users into an undesirable trade-off between utility and privacy.

In the realm of public safety and trust, the use of ML in surveillance can infringe upon individuals' rights and amplify concerns about abuse of power. Furthermore, ML algorithms used in decision-making processes can perpetuate bias and discrimination, affecting the fairness and justice of outcomes. Finally, the spread of misinformation through AI-generated content challenges the integrity of free

speech and trust in digital media. These examples emphasize the need to develop frameworks that ensure a responsible deployment of these technologies and trustworthy user-machine interaction.

In this thesis, we *examine the risks posed by ML algorithms and develop robust systems to mitigate these risks and facilitate a trustworthy interaction, ultimately empowering users with greater control over their data*. Specifically, we have explored different ubiquitous ML applications encompassing various data modalities such as speech and vision. For each application, we define the privacy and security risks users face when interacting with the technology, highlight the shortcomings of the existing deployments, and propose a system that enhances the utility and privacy and integrity trade-offs and enables a more trustworthy interaction.

1.1 Voice-based User-Machine Interaction

Motivation: Speech is a natural form of human communication which makes it a very convenient vehicle for human-computer interactions as well. Scientists have been developing systems for natural speech understanding for decades. This task has become much easier in recent years due to the massive growth of smart devices, the advancement of machine learning algorithms, and the abundance of public speech data. Thanks to these advancements, machines can now understand speech, generate close-to-natural speech, differentiate between different speakers, and even detect many paralinguistic features about the speakers such as their emotions, health condition, age, gender, and mental health.

Speech technologies such as automatic speech recognition, speaker identification, keyword spotting, and sound classification have become very reliable. Cloud operators offer these technologies as a machine learning as a service (MLaaS) business model. These services enable the integration of such technologies in many appliances and devices that we interact with in our daily lives. For example, automatic speech recognition is capable of accurately transcribing long conversations in a few seconds for a reasonable cost. This technology is highly valuable in many domains such as journalism, hybrid meetings, live captions for videos and conferences, customer service calls, and online education.

Another groundbreaking application is voice-activated devices such as standalone voice assistants, e.g. Google Home and Amazon's Echo, and built-in voice-

enabled devices such as smart appliances and IoT devices. Voice-activated devices are increasingly pervasive in households. As of 2019, about 35% of U.S. households are equipped with at least one voice assistant, and this rate is expected to increase to about 75% by 2025 [2]. By 2024, it is predicted that the number of voice-activated devices will reach 8.4 billion units; this number is higher than the world's population [3]. Voice-activated devices offer their users a convenient way to access information, set alarms, play games, or control appliances, especially when traditional (physical) I/O modalities are inconvenient.

These technologies are developed with performance and user experience as their main driving objectives. However, they are accompanied by unprecedented privacy, security, and integrity threats that have become more prevalent with their wide deployment. These threats include cloud access to private recordings, unauthorized voice activations of smart speakers, unauthorized voice biometrics collection, and speaker impersonation. Recent privacy regulations, such as the GDPR and CCPA, provide guidelines for protecting users' privacy. However, current technologies and cloud services fall behind in meeting these requirements. In this thesis, we analyze the emerging privacy and security threats accompanying speech technologies. We develop practical systems to mitigate the risks while preserving the utility and convenience of the current technology. We validate the efficacy of our work by empirical and theoretical analyses.

System Model

Fig. 1.1 shows a general pipeline of a human-machine speech interaction. The figure illustrates the user's interaction with a voice-enabled device, such as a home assistant or a smart vehicle assistant. The pipeline consists of an input speech sensor (microphone) and three main voice-related ML components:

1. A keyword spotting system (KWS) that activates the device when the user says the wake-up (activation) phrase. This system runs locally on the device and only communicates with the cloud when it detects a possible activation.

2. A speaker identification system that identifies the speaker from their voiceprint. It can run locally or on the cloud, and it facilitates two main objectives: authentication and personalization.

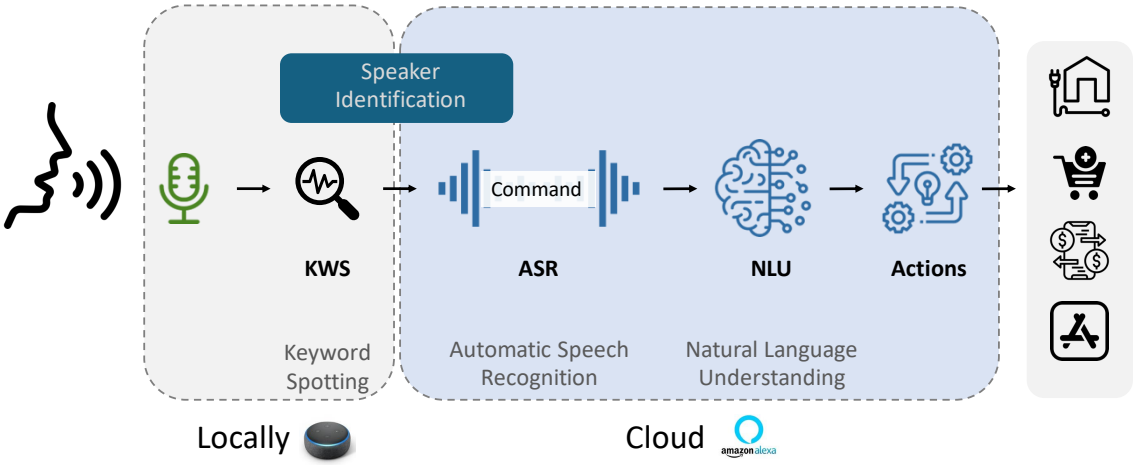


Figure 1.1: A general pipeline of human interaction with a voice-enabled device. The pipeline consists of three main voice-related ML components: (1) a keyword spotting system (KWS), (2) a speaker identification system, and (3) an automatic speech recognition system (ASR), along with the natural language understanding unit (NLU) that controls the device's action.

3. An automatic speech recognition system (ASR) that converts the recorded speech into a transcript. ASR usually runs on the cloud because it is a computationally expensive task, as explained in Chapter 3.

The voice-enabled device is equipped with microphones that are always listening for a wake-up word followed by a voice command and are connected to the cloud for command transcription and execution. Once the keyword spotting module detects the activation keyword, it activates the pipeline. The speaker recognition module detects the speaker's identity for authentication and to facilitate a personalized experience. Then, the speech command is sent to the cloud for transcription and execution. Finally, many natural language tasks can be performed on the transcript to further understand the user's speech or request. Next, we detail the threat model accompanying each technology.

Threat Model

Keyword Spotting

Voice Assistants (VA) offer their users a hands-free natural interaction via voice which is preferable in many situations such as driving, cooking, and exercising. VAs are equipped with microphones that are always listening for a wake-up word

followed by a voice command and are connected to the cloud for command transcription and execution. Thus, despite their convenience, they introduce crucial security and privacy implications [4]. It is unprecedented to have an *always* listening device in our most private and intimate spaces, potentially recording private conversations and sending them to the cloud [5]. Moreover, voice assistants collect sensitive information like credit card data and bank accounts, and control physical smart home devices such as the lights, garage door, and other smart appliances. Compromising the VA by a malicious activation gives the adversary access to all the powerful controls the VA possesses. Adding to the problem is the disconnect between the VA's behavior and the user's expectations. Thus, the integrity of the VA operation is a critical requirement for a safe and secure deployment that matches the user's expectations.

Automatic Speech Recognition.

Cloud providers, such as Google, Amazon, and Microsoft, offer online APIs for automatic speech recognition (ASR), which achieve a near-human performance, especially in low-noise settings [6]. This service enables near real-time transcription with high accuracy and minimal computation burden on the customers. However, the accessibility and scalability of this technology come at the cost of customers' privacy. Speech is a rich source of personally identifying information. The acoustic features are biometric identifiers of the speakers, enabling speaker identification from short segments of speech [7]. Much paralinguistic information such as age, sex, accent, health condition, and emotional state [8] can be inferred from speech recordings. The linguistic and textual content also conveys lots of sensitive information [9]. For example, medical recordings can contain private health information about patients [10], and business meeting recordings can include proprietary information. The same applies to journalism, educational, and legal settings. The privacy of the speakers in most of these settings falls under federal laws such as FERPA and HIPAA. Moreover, current cloud services already support several speech processing APIs like speaker identification, and text analysis like topic modeling, document categorization, sentiment analysis, and entity detection that can automatically extract sensitive information from speech and its transcript. The unregulated usage of speech recognition APIs can significantly undermine the privacy of their customers.

Speaker Identification

Users' voice biometric features contain a voiceprint that can be used to identify and authenticate the speaker. Aside from voice assistants, this technology has been deployed in real-world security-sensitive applications such as phone banking services (e.g. HSBC [11] and Chase [12] banks). Voice authentication, like other biometric-based authentication schemes, is more convenient to the user than conventional authentication mechanisms such as passwords, security questions, or private keys. However, voice authentication is vulnerable to impersonation and presentation attacks. Speech synthesis and voice conversion [7] are two technologies that can generate synthetic speech in a target speaker's voice. These technologies jeopardize the security of applications that rely on speaker identification as an access mechanism. Moreover, speaker identification can be used for unauthorized surveillance. With the abundance of speech data on social media platforms, people can be enrolled in a speaker identification system without their consent for tracking and surveillance. To mitigate these threats, we need to verify that the speaker identification request is not synthetic and is intentionally initiated by the corresponding user.

1.2 Vision-based User-Machine Interaction

Vision-based user-machine interactions represent a rapidly evolving field within the realm of AI and human-computer interaction. This technology involves using visual input, usually from a camera, to enable machines to identify humans and respond to their actions or presence. This type of interaction goes beyond traditional input methods like keyboard and mouse, offering a more intuitive and natural way for users to engage with technology. These technologies have found significant applications in education, retail, healthcare, public safety, and social media platforms. In retail, for instance, vision-based customer identification can enhance customer experience by providing interactive and personalized shopping experiences. In healthcare, vision-based systems could monitor patients' physical responses and facial expressions, providing valuable data for diagnostics and treatment. Vision technologies on social media platforms have transformed the way users interact with content and each other. Platforms offer many vision services for the users' uploaded visual content such as facial recognition for tagging friends

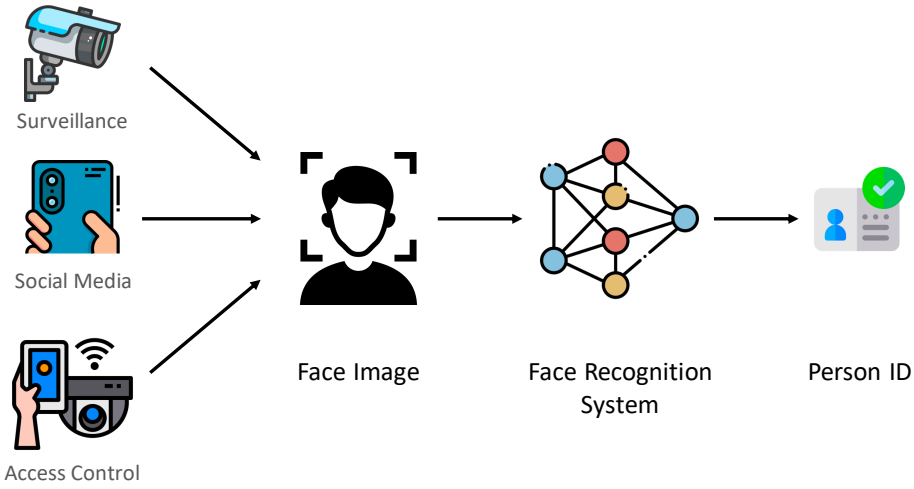


Figure 1.2: Face recognition pipeline. Its applications include surveillance, social media, and camera-based access control. The application captures an image of the user's face, feeds it to the face recognition ML system, and gets the user's ID.

in photos, and applying filters to enhance images and video quality. Moreover, such technologies can detect personal attributes about the customers such as age and gender to personalize their experience and automatically suggest relevant and possibly addictive content.

This seamless and personalized approach naturally leads to the focus on face recognition technology, a key component in vision-based interactions. Face recognition, with its ability to identify and verify individuals based on their facial features, is a valuable tool in ensuring security and personalization. Its application ranges from unlocking personal devices like smartphones and laptops to enhancing security measures in public spaces and airports. However, the deployment of face recognition technology raises significant privacy and ethical concerns. Issues related to consent, data security, and the potential for misuse or bias in the algorithms are critical. Moreover, the accuracy and fairness of face recognition systems are under scrutiny, especially in cases where they have been found to exhibit biases against certain demographic groups. The ongoing development and application of face recognition technology thus present a critical intersection of technical innovation and ethical responsibility, necessitating a careful and balanced approach to its use and regulation.

System Model

Fig. 1.2 shows an example of the face recognition technology. The system captures an image of the user's face, feeds the image to the face recognition model, and gets the user's ID. Typically, the face recognition model functions as a metric embedding network, which transforms the input image into a representation within an embedding space. This network is designed and trained to position images that are semantically similar in close proximity to each other within this space.

Threat Model

Race and gender bias in facial recognition has been studied extensively over the years [13, 14, 15, 16, 17]. It is widely reported that face recognition systems can be biased towards male and light-skinned faces while demonstrating the lowest accuracy on dark-skinned female faces [18, 17, 16]. Other facial semantics like age [19, 16, 20], pose [21] and hair [20, 22] have also been shown to contribute to facial recognition performance.

The prominent natural datasets used to train face recognition models include LFW [23], CASIA WebFace [24], VGGFace [25, 26], and Flickr-Faces-HQ [27]. These datasets are sourced from the web, mostly contain celebrity faces, and can be biased towards certain demographics [28]. Many balanced datasets have been proposed [29, 30, 31] to overcome problems with the above datasets. However, even a demographically balanced dataset can show disparate facial demographic performance [32] due to limiting factors like lightning, pose, and image quality. In this thesis, we aim to synthesize a balanced dataset representing different demographics with ground truth labels of different semantic attributes for each identity in the dataset.

1.3 Thesis Contributions

In this thesis, we design tools and systems to address the threats facing various speech and vision technologies, which billions of users interact with daily.

EKOS **[33].** First, we assess the privacy risks associated with false activations of voice assistants (VAs) like Amazon Echo. To address these risks, we propose

EKOS: an ensemble of smart devices. EKOS leverages the diversity of physical channels, the nature of speech signals, and the diversity of ML architectures to enhance the robustness of the keyword spotting system against false activation threats.

Preech **[34].** Second, we investigate the privacy concerns associated with cloud-operated speech transcription services. While these services offer high utility, they process and store users' voice biometric data and transcribed content in the Cloud, potentially jeopardizing privacy. To address these concerns, we propose Preech, an end-to-end system that applies voice conversion and differential privacy to protect speakers' voice biometrics and textual content.

Mystique **[35].** Third, in the context of speaker authentication, we argue against relying solely on voice biometrics for security-critical applications, such as voice banking. Through physical demonstrations, we design an impersonation attack using physical objects, such as a structure of tubes, to reshape a speaker's voice and successfully fool a speaker identification system. This chapter highlights the need for multi-factor authentication approaches that consider the vulnerabilities of voice biometrics.

Visual Semantic Robustness [36]. Finally, in the context of face recognition, we utilize Text-to-Image diffusion models to synthesize a diverse face image dataset. We instruct the model to generate images representing various demographic groups and incorporate a semantic guidance network that steers the diffusion model's generation toward incorporating a variety of semantic attributes. We further validate the faithfulness of the generated dataset using user surveys. We believe this dataset will be very useful for future research assessing the performance of face recognition models.

Chapter 2

EKOS: Ensemble for Robust KeywOrd Spotting

2.1 Introduction

Voice assistants (VAs) interpret voice commands from their users to assist in different tasks, access services, and control smart devices. A typical voice assistant continuously samples audio through its microphone to detect a user saying a keyword, such as "Alexa," "Siri," or "Google." This process, referred to as Keyword Spotting (KWS), serves as the primary access control to an active voice assistant. Once it detects the wake keyword, the voice assistant streams the subsequently recorded audio to be analyzed as a voice command.

The Keyword spotting (KWS) task is a two-stage process spanning the device and cloud: a local on-device model first detects the keyword and sends a speech segment to the cloud, which verifies the keyword and processes the accompanying command [37]. Verification is necessary since on-device models are typically less accurate; they are optimized to minimize their compute footprint and latency of predictions [38, 39, 40], whereas the cloud model can be a full-fledged natural language model with higher precision.

In this chapter, we find that unauthorized accidental activations due to poor precision of the on-device KWS model can lead to significant privacy violations with up to a minute of private speech being uploaded to the cloud. In addition, adversaries who wish to get unauthorized access to the private VA may systematically trigger such unauthorized activations with adversarial examples. This

adversarial activation puts the device integrity and the user's security at risk, given the numerous appliances and services connected to voice assistants (e.g., garage door, lights, and credit cards) [41, 42, 43].

As the entry point for any interaction with the VA, improving the precision of on-device KWS directly limits the extent of private conversations leaked to the cloud and reduces the attack surface available to adversaries. Existing defenses to these problems rely on generic machine learning approaches, such as adversarial training [44]. Such approaches typically harm the natural accuracy—an unacceptable proposition for VAs—or fail to provably increase the cost of an adversary launching an over-the-air attack. Other approaches employ liveness detection mechanisms [45] that potentially introduce additional privacy problems and do not address the accidental activations problem. In short, this chapter considers the question of *how to improve the robustness of KWS against accidental and adversarial activations while preserving its precision?*

In this chapter, we design, implement, and evaluate EKOS (Ensemble for KeywOrd Spotting) as an affirmative answer to the above question. EKOS leverages the semantics of the KWS task to arrive at a more favorable tradeoff between the robustness and precision of the KWS model. First, EKOS incorporates spatial diversity from the acoustic environment at both training and inference time to minimize distribution drifts responsible for accidental activations. Second, it exploits a physical property of speech—its spectrum redundancy—to deploy an ensemble of models trained on different harmonics. It provably forces the adversary to modify more of the frequency spectrum to obtain successful adversarial examples.

Modeling distribution drifts responsible for accidental activations is challenging because the physical environment evolves constantly. EKOS addresses this issue by exploiting the natural randomness from the physical environment (such as room impulse responses) and ensembling other voice-aware devices available in the vicinity of the virtual assistant. In particular, EKOS performs KWS with an *ensemble* of models, each served by a device with varying internal sensors, hardware, and channel from the user. EKOS uses the *diversity* ensuing from the ubiquity of smart devices in a given environment, such as tablets, computers, and smartphones, to improve the precision of the KWS task by combining the detection results from these devices.

Improving robustness to adversaries is more challenging because they can still overcome ensembles of models [46, 47], especially when the feature space is

common to all models. EKOS addresses this challenge by utilizing the redundancies in speech signals and properties of the KWS task. A speech signal carries replicas of the same content (i.e., a word) at different frequency components: harmonics. It is thus possible to slice the signal's spectrogram into different slices and assign each slice to a different model without much impact on the natural accuracy. We design these slices and architectures to exhibit poor transferability. Further, EKOS randomizes the slice-architecture combinations in the ensemble at run-time. This approach increases the cost of an adversary because they now have to perturb a majority of the frequency slices before they can control the predictions of the ensemble.

In summary, our contributions are as follows:

1. We show that privacy leakage is greater than previously believed when on-device models send private conversations to the cloud due to accidental activations. Previous analysis [48] reported misactivations resulting in 10 seconds of speech being leaked; our evaluation shows that some misactivations lead to up to a minute of speech leaking to the cloud (Sec. 2.8 — Fig. 2.7).

2. We design an ensemble of KWS detectors that can run on distributed devices in an environment. This ensemble leverages the semantics of the KWS task, the properties of the audio channel, and the nature of the speech signal to introduce real diversity to the prediction task (Sec. 2.5).

3. Our end-to-end evaluation shows that an ensemble of three to five devices, with random slicing and architectures, increases the cost of adversarial attacks (Sec. 2.7, 2.8). At the same time, EKOS preserves the natural accuracy, approximating the baseline accuracy and has little performance overhead (Fig. 2.3, 2.5). We validate the performance of EKOS with over-the-air experiments on commercial devices; we find that EKOS improves the precision of the KWS task in non-adversarial settings (Sec. 2.7, 2.8).

4. We generate and release[1] a dataset of the Amazon Echo's wake keywords: {Alexa, Computer, Amazon, Echo}. We use this dataset to validate EKOS robustness on Amazon's Echo devices. The same methodology can be followed for other commercial devices and keywords.

[1] https://github.com/wi-pi/EKOS

2.2 Background on Keyword Spotting

The KWS task is responsible for detecting a set of predefined *keywords* in an audio stream. Typically, the VA's microphone(s) capture the over-the-air audio stream. Then, the VA performs audio pre-processing and KWS classification.

Physical Environment. When an audio signal is transmitted over-the-air, the signal reflects off the room walls and the objects in the room. The received signal at a microphone is the sum of the line-of-sight and reflected audio copies, known as reverberations or echo, as shown in Fig. 2.1. The reverberation can be modeled via a room impulse response (RIR) $h(t)$, and the received signal is the convolution of the transmitted audio and the RIR, $r(t) = s(t) * h(t)$, where $h(t)$ depends on the speaker and microphone locations, the room dimensions, objects, and the materials absorption factors. Hence, $h(t)$ is unique per every room and speaker-microphone setup.

Feature Extraction. The mel-frequency cepstrum coefficients (MFCC) are the conventional features used for speech recognition tasks including ASR and KWS; they reduce the dimensionality of an audio signal, $r(t)$, to a 2D temporal-spectral map. The MFCCs are computed as follows [49]: (1) divide $r(t)$ into short time frames (20–40ms); (2) compute the short-time Fourier transform (STFT) of these frames; (3) map the STFT linear frequency scale to the mel-scale using a mel-spaced filterbank. The mel-scale approximates the human auditory system as it applies more (fewer) filters in the low (high)-frequency range; (4) take the log of the power; and (5) apply the discrete cosine transform (DCT). The MFCCs are the coefficients of the resultant spectrum at each time frame.

Classification. The KWS task employs a multi-class model $f(\cdot)$ to classify an input audio $r(t)$ as a label corresponding to the detected keyword, with the "*unknown*" label for non-keyword speech. The model consists of three components: (1) extracting MFCC features from $r(t)$, (2) feeding the MFCCs to a deep neural network (DNN), and (3) computing an average score of the individual frames' posterior scores to report the keyword score. Earlier research on KWS considered DNN architectures which treated MFCCs as 2D features [50, 51].

Choi et al. [52] were the first to treat the MFCCs as a 1D time signal, where the frequency coefficients are the input channels. They proposed TC-ResNet, a

temporal convolution residual network architecture. The 1D temporal convolution reduces the feature map size and has a large receptive field since the filter covers the whole range of frequencies (channels). It achieves better performance at a smaller number of parameters and computations, hence, lower latency. We utilize these architectures in the design of EKOS.

2.3 Background on KWS Misactivations

The KWS performance is crucial for the VA's user experience [4]. A near-optimal true-positive rate is essential for the device's responsiveness and utility. On the other hand, a KWS *misactivation* compromises the user's privacy and the VA's integrity. A misactivation takes place when the VA is activated by an unauthorized command, i.e., a sound that is not the correct keyword. In this chapter, we consider two types of misactivations: accidental and adversarial activations.

Accidental Activations

An *accidental* activation happens when the KWS model *mistakenly* interprets a sound that is not the keyword as a positive activation, i.e., a false-positive detection. In such a case, the VA inadvertently records the user's private conversations and sends them to the cloud for transcription and execution.

The privacy threats stemming from having an *always* listening microphone in private spaces have been extensively studied [53, 54, 55, 56, 57, 58, 5]. Recently, two studies [48, 37] performed a comprehensive analysis of the accidental activation triggers on a variety of VA devices and keywords. They use TV shows, newscasts, and speech datasets to locate phrases that accidentally trigger each VA. Dubois et al. [48] observed 0.95 misactivations per hour, where they identified some activations lasting for at least 10 seconds. Likewise, Schönherr et al. [37] located hundreds of accidental activations in the evaluated media. They observe that the cloud-based KWS verification model reduces the number of local misactivations. Yet, more than half of the evaluated triggers still incorrectly activate the cloud's model. Moreover, they created a dataset of more than 1000 English n-gram phrases that are phonetically similar to the commercial keywords; these phrases are likely to cause misactivations. Both studies noted that the VA's operation is non-deterministic; it is hard to predict when a device may be accidentally activated.

Adversarial Activations

As far as their integrity[2] is concerned, KWS models are vulnerable to inference time adversarial examples [60, 61], where an adversary constructs *imperceptible* commands hidden in a non-suspicious audio utterance, such as music or a YouTube video, to wake up and interact with the VA [42, 41, 62].

Given an audio signal $r(t)$, and a KWS model $F(\cdot)$, the attacker's objective is to find a small perturbation δ, such that $F(r(t)+\delta) = y$, where y is the target keyword that triggers the VA. We refer to this attack as an *adversarial* activation.

Adversarial Examples on Audio. Carlini and Wagner [63] constructed a targeted white-box attack on the neural ASR system, Deep Speech. The attack is *digital*; i.e., it does not consider a physical channel and assumes the audio stream is directly fed to the model. The attack optimizes this objective:

$$\min \ \ell(F(s+\delta),\ y) \ + \ \alpha \cdot \|\delta\|_\infty \quad \text{s.t.} \quad \|\delta\|_\infty < \epsilon, \tag{2.1}$$

where s is the input to the neural network $f(\cdot)$, δ is the perturbation, y is the target label, ℓ is the loss function, ϵ is the attack budget which bounds the maximum added perturbation, and α is a hyperparameter; the adversarial example is $s'(t) = s(t)+\delta$. The authors choose ℓ to be the CTC (Connectionist temporal classification) loss and use the max-norm ($\|\cdot\|_\infty$) which has the effect of adding a small perturbation consistently throughout the utterance samples. This attack, however, is against ASR, not KWS; both ASR and KWS have similar preprocessing pipelines involving MFCCs, but the task solved by each model is different.

The adversarial example $s'(t)$ constructed with Eqn. 2.1 is neither completely imperceptible nor effective over-the-air. The former requires that $s'(t)$ sounds very similar to $s(t)$ to a human listener. The latter requires that $F(s'(t) * h) = y$ for any h, where h is the physical environment room impulse response (RIR) (Sec. 2.2). Following this initial attack, recent works have focused on solving these two challenges.

Imperceptibility. Schönherr et al. [62] examine a different bound on the perturbation that better addresses the human auditory system perception. They propose

[2]We note that integrity is not the only property adversaries may target. Attackers also jeopardize the availability of the ML system, as shown in recent work on the presence of adversarial music [43] or Sponge Examples [59].

psychoacoustic masking, as in MP3 encoding, to hide the perturbations around the original speech frequency components, where they are barely perceptible to humans. However, their attack assumed a perfect channel; i.e., it is not robust over-the-air.

Over-the-air Robustness. Adversarial examples are not robust in the physical world when the input signal is subject to environmental variations (transformations)—as initially observed in vision [64]. The adversary can adapt by considering the distribution of possible transformations, and optimizing the perturbation over the Expectation over Transformation (EoT) [64], such that the resulting perturbation transfers across these transformations *on average*. Qin et al. [42] and Schönherr et al. [62] apply EoT to the acoustic domain to capture room reverberation. They convolve the audio signal with RIR:

$$\min \underset{h\sim\mathcal{H}}{\mathbb{E}}[\ell(F((s+\delta)*h),\ y)]\ +\ \alpha\cdot\|\delta\|_p \quad \text{s.t.} \quad \|\delta\|_p < \epsilon, \tag{2.2}$$

where $\mathcal{H}$ is the RIR distribution of the possible room dimensions, and speaker and microphone locations.

2.4 System and Threat Models

System Model. We assume the VA to exist in an environment that contains a set of trusted devices, such as smartphones, computers, and tablets. Each device has at least one microphone, a network interface, and computing capabilities. We believe these assumptions are realistic about the households or spaces with a VA. As in any realistic setting, these devices are randomly located within the environment, experiencing random acoustic channels, and have inherent hardware variations, as shown in the setup at Fig. 2.1 (left). The user deploys EKOS by installing an app on their microphone-equipped devices. The app runs in the background, reads the microphone, performs KWS, and communicates with the VA.

Threat Model. We consider two independent threat vectors that result from false VA activations due to the KWS model's imperfections. Both vectors are different in the adversary definition, attack implementation, and the subsequent privacy and security violations. We do not suggest that the same adversary can execute both

threat vectors; yet, both threats are enabled by the same vulnerability: a false VA activation.

The first threat vector covers a *remote* and *passive* adversary with access to the VA's recordings once they are uploaded to the cloud. Because of imperfections of KWS models, the VA can be accidentally triggered, causing it to record conversations not intended as commands. Although the cloud has access to the users' legitimate commands, accidental activation poses real privacy threat [37, 4]. Under a legitimate activation, the user is aware that their commands will be recorded and uploaded to the cloud. Detecting the legitimate keyword forms an implicit consent to be recorded. However, in the case of accidental activation, the recorded conversations are private; the users are unaware and did not approve the recording. The privacy concerns stem from the content of the private conversation, the context, and the background noise. Under this setting, the user's privacy can be compromised in different ways: (1) the cloud uses these recordings to train ML models [65, 66], these models can memorize the training data [67]; (2) an adversary compromises the cloud servers and leaks such conversations [68, 69, 70]; or (3) third-party transcription contractors or law enforcement agencies can potentially have access to the private recordings [71, 72, 73].

The second threat vector covers a *remote* and *active* adversary who activates the VA with imperceptible perturbations hidden in a non-suspicious audio utterance, e.g., music. This adversary can remotely trick the user into playing audio from a TV, YouTube, or SoundCloud, which embeds the imperceptible perturbation – scaling the attack to many users. Prior research has demonstrated the feasibility of generating adversarial samples in the form of inconspicuous background music [43, 41]. Once the VA is activated, the adversary can push commands to activate malicious skills or interact with physical devices in the user's environment. Such adversarial activation puts the device's integrity and user's security at risk given the numerous services and appliances connected to the VA (e.g., garage door, bank accounts). We consider a white-box attacker who has access to the KWS model parameters as well as EKOS's setup internals. This adversary can launch adaptive attacks in an attempt to circumvent EKOS. Note that the adversary has no *physical* access to the VA; otherwise, the adversary can interact with the device using their own voice without the need to launch adversarial perturbations.

Threat vectors that directly attack the microphone interface, such as ultrasound [74, 75] and laser attacks [76], are outside the scope of this chapter as they are not

based on false activations of the VA. Our work is orthogonal and can compose well with approaches to defeat these other threats [77, 78]. In Sec. 2.9, we discuss how EKOS can address these threats.

2.5 EKOS: Ensemble for KeywOrd Spotting

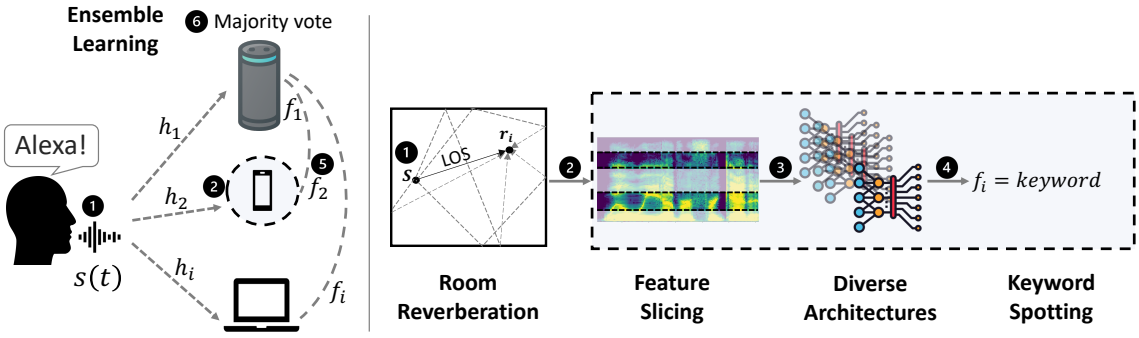


Figure 2.1: EKOS Overview. Left: High-level operation. Right: Details of the signal processing pipeline at the device. Step (1) is the spoken speech signal, step (2) is the received signal after experiencing the acoustic channel, step (3) is applying a random feature slicing filter, step (4) is passing the filtered signal through a randomly chosen architecture, step (5) refers to sending the decisions from individual devices to the VA, and step (6) is the final ensemble output of a majority vote.

High-level Overview

EKOS comprises two components: a machine learning-based component (ensemble learning) to improve the robustness of the KWS task against accidental activations, and a signal processing-based component (feature slicing) to handle adversarial examples against KWS.

Fig. 2.1 illustrates the high-level operation (left) and the processing pipeline of EKOS (right) at each device. EKOS deploys *diverse* keyword spotting models on a set of commodity devices, such as smartphones, smart TVs, laptops, and edge devices, and combines their decisions to improve the overall classification performance, a technique known as ensemble learning. EKOS views the output of each model as an independent random variable (vote) specifying the identified keyword from the input audio. All devices run a lightweight webserver and are

connected to the same wireless network. The ensemble integration happens as follows: (1) the main VA (server) listens continuously; it initiates the KWS vote collection from the other devices (clients) upon detecting a keyword; (2) the main VA issues parallel requests to the client devices and waits for their response; (3) each client device buffers its microphone signal and waits for an inference request; (4) upon receiving one, it runs its KWS model and returns the predicted hard label; and (5) the main VA outputs the final prediction through a majority vote mechanism.

Accidental Activations. A key point to harvest the gain of ensemble learning is to ensure (as much as possible) that the models' errors are uncorrelated [79, 80]; a voting mechanism in such a case would reduce the false positive rate responsible for accidental activations. EKOS satisfies this condition by introducing different levels of diversity at the model design and the received input signal.

EKOS processes speech samples using a set of l KWS models. We introduce diversity into the models decisions by selecting different architectures and hyper-parameters for each model (Sec. 2.5). EKOS allows this KWS ensemble to be either centralized in the VA or distributed over a set of N smart devices existing in the environment. These devices experience different acoustic propagation channels and have inherent hardware diversity. Hence, they capture uncorrelated samples of the audio stream [81].

Adversarial Examples. In its second component, EKOS leverages diversity from the feature space and the environment to increase the cost of generating adversarial examples against KWS. EKOS decomposes the speech spectrum into a set of possibly overlapping spectrum slices (feature slicing in Sec. 2.5). Because of the nature of the speech signal, the spectrum slices contain harmonics that encode replicas of the speech content. These frequency components, however, undergo different transformations as they travel across the physical channel. As a result, each spectrum slice is useful in identifying keywords found in speech but requires the adversary to inject a perturbation specific to this spectrum slice.

Ensemble devices behave independently at run-time; each device chooses a subset of spectrum slices at random. Then, it passes the slice into a randomly chosen architecture (randomized ensemble in Sec. 2.5). Each model assigns the slice with a classified keyword and relays its label to the VA.

The robustness in this approach arises from three insights related to the classification of audio signals. First, the channels are spatially independent; the adversary has to account for more transformations in generating the adversarial examples. As specified in Sec. 2.3, an adversary uses the expectation over transformation technique to generate adversarial examples that are adaptive to this defense, where each transformation represents a simulated channel. Having a set of simultaneous independent channels constrains the attacker's optimization problem further; the result is a less optimal (larger) perturbation (Sec. 2.7). Second, adversarial examples in the audio domain have poor transferability properties; an adversarial example optimized for a slice-architecture combination does not transfer easily to other combinations. This insight is supported by previous results about the transferability of audio adversarial examples [82], as well as our results presented later in Sec. 2.6 and Fig. 7.3. Third, EKOS chooses the slices and architectures randomly at run-time, which forces the adversary to cover more slice-architecture combinations to ensure a sufficiently large probability of attack success. In the following, we discuss EKOS's feature slicing and randomized ensemble.

Feature Slicing

The feature slicing in EKOS applies bandpass filters to the speech spectrogram to select frequency slices. EKOS leverages a key property of audio signals: they carry replicated information across the different frequency bands. This information content, however, is not uniform; bands in lower frequencies contain more information than bands in the higher frequency range. This insight forms the basis for the MFCCs, which perform non-linear mel-scaling of bands as inspired by the human auditory system [49]. In EKOS, we follow a similar methodology; we define six bandpass filters, three at the lower end of the spectrum spanning the bands: (1)-[0Hz, 750Hz], (2)-[700Hz, 1700Hz], (3)-[1650Hz, 2900Hz], and another three at the higher end of the spectrum: (4)-[2850Hz, 4350Hz], (5)-[4300Hz, 6050Hz], (6)-[6000Hz, 8000Hz]. Notice that the bandwidth of the filters increases linearly from 750Hz to 2000Hz with 250Hz increments. These bandpass filters are the building blocks of the feature slicing filters.

The design of these filters involves two tradeoffs between natural and adversarial robustness. The first tradeoff is the width of the filter. A set of narrow filters force the attacker to add the perturbation in more concentrated frequency regions,

making it harder to hide imperceptible perturbations [42] – at the cost of reduced natural accuracy. The bandwidth has to be wide to capture more content of the speech signal.

The second tradeoff concerns the overlap between the filters. If filters overlap to the extent that they all share a common frequency band, the attacker's strategy would be to target this single shared band, resulting in a single perturbation that transfers across all the filters; the attacker's optimization function resolves to a single objective as in Eqn. 2.3. On the other hand, if the filters have no overlap, the number of possible filters will be limited. Moreover, the attacker can target such mutually exclusive bands separately, where the final perturbation is the sum of the individual perturbations. Hence, it leads to less robustness and randomness in the EKOS ensemble.

As such, we design the set of filters G such that each feature slicing filter $g \in G$ includes two bandpass filters, one chosen from the set of lower bands (filters 1, 2, and 3) and the other chosen from the set of higher bands (filters 4, 5, and 6). This design results in G comprising nine combinations $\{(1,4),(1,5),(1,6),(2,4),(2,5),(2,6),(3,4),(3,5),(3,6)\}$. Any single band is repeated only three times across the filters set G. Hence, $g = w[s_l, e_l] + w[s_h, e_h]$, where w is a rectangular window function, s_l, e_l are the low-frequency window start and end frequencies, and same for s_h, e_h for the high-frequency window. This design balances the amount of information passed by each slicing filter and intentionally adds overlap between the filters without sharing any single band among all of them. We show later (Sec. 2.7) that the designed filters preserve the model's natural accuracy and provide feature space diversity such that their ensemble accuracy approximates the baseline accuracy.

Runtime Ensemble

In an environment with N devices, the user's speech signal $s(t)$ travels over a set of channels $h_i(t)$; each device d_i receives a signal $r_i(t) = s(t) * h_i(t)$. A device d_i has access to the set of G filters as defined in Sec. 2.5 and a set $\mathcal{F} = \{F_k | \, 1 \leqslant k \leqslant K\}$ of architectures, where K is the number of baseline KWS architectures. We refer to $F_{j,k}$ as the architecture F_k trained after applying filter g_j. Each device can run one or a subset of KWS models simultaneously based on its processing capabilities, the availability of other devices, and the user's preferred level of privacy and utility

(Sec. 2.9). The device chooses randomly a subset $G_i \subseteq G$ frequency filters. It applies each $g_{i,j} \in G_i$ to r_i resulting in a set of signals $r_{i,j}(t) = r_i(t) * g_{i,j}(t)$. Then, the device assigns each $r_{i,j}(t)$ a random architecture $F_k \in \mathcal{F}$; each model outputs $f_{i,j} = F_{j,k}(r_{i,j}(t))$, where $f_{i,j}$ indicates the output class (keyword). Each device d_i sends the set $\{f_{i,j} | 1 \leqslant j \leqslant |G_i|\}$ to the VA for the final decision. The VA receives a set of l decisions from all the devices, such that $l = \sum_i^N |G_i|$. It performs majority voting by choosing the class with the highest number of votes.

We exhaustively searched through the models trained over slice-architecture combinations to ensure adversarial examples have low transferability. Fig. 7.3, in the Appendix, shows that these models exhibit poor transferability. We conjecture that reducing the overlap between the filters in G contributes to this observation. This poor transferability is an important property for EKOS's robustness, as discussed in Sec. 2.5.

An adaptive attack can target the ensemble models simultaneously [47, 46] given a higher perturbation budget. Thus, we introduce inference time randomization to EKOS's operation: we randomize the slice-architecture combination. At each T_i interval, each device i randomly selects a frequency filter subset $G_i \subseteq G$ and assigns each filter $g_{i,j} \in G_i$ a random architecture $F_k \in \mathcal{F}$, where T_i is independently set by each device. Hence, the slice-architecture combinations independently and randomly change every T_i.

Finally, EKOS design is flexible and can be optimized towards a customized utility-robustness level. The user has the option not to apply the feature slicing prior to the ensemble. In such a case, EKOS does not apply the randomized feature and architecture selection. It just passes the received signal at each device to a model F_i and aggregates the decisions at the VA. This mode improves the KWS accuracy against accidental activations but not against adversarial activations. Moreover, the user sets EKOS' hyperparameters, such as N, l, K, and $|G_i|$, to optimize the computational overhead (Sec. 2.9).

Robustness Properties

The robustness of EKOS arises from the increase in the attacker's cost. The original attack requires optimizing over a single constraint to force a label y, such that:

$$\min ||\delta||_p, \quad \text{s.t.} \underset{h \sim \mathcal{H}}{\mathbb{E}} [F((s(t) + \delta) * h(t) * g(t))] = y, \tag{2.3}$$

where $h \sim \mathcal{H}$ is a random variable describing the channel between the speaker and possible devices.

Introducing the ensemble of slice-architecture combinations, and assuming the attacker knows the chosen slices and architectures, the attacker's optimization objective comprises multiple constraints. Without loss of generality, assume that each device d_i runs a single model $F_{j,k}$ for a specific filter $g_{i,j}$. The attacker's objective can be represented as:

$$\begin{gathered} \min \|\delta\|_p \text{ s.t.} \\ \Bigg(\mathop{\mathbb{E}}_{h_0 \sim \mathcal{H}} \left[F_{j,k} \left((s(t) + \delta) * h_0(t) * g_{0,j}(t) \right) \right] = y \\ \dots \\ \wedge \mathop{\mathbb{E}}_{h_{l/2} \sim \mathcal{H}} \left[F_{j,k} \left((s(t) + \delta) * h_{l/2}(t) * g_{l/2,j}(t) \right) \right] = y \Bigg). \end{gathered} \tag{2.4}$$

Because of majority voting, the attacker has to satisfy a set of $l/2 + 1$ constraints to control the ensemble output. Intuitively, this optimization problem is more constrained and will result in a larger perturbation compared to the less constrained problem of one slice-architecture combination. This property, however, only holds when gradients of the constraints are linearly independent. Otherwise, the same perturbation may be able to force models trained on two or more spectrum slices to misclassify when these models' gradients are linearly dependent. In EKOS, we encourage gradients to be linearly independent with diverse architectures and by designing the filters to have little overlap (Sec. 2.5).

EKOS randomizes the slice-architecture selections at run-time to increase the cost of the attack. Given a set of M possible channel-slice-architecture combinations, the adversary has to attack the M combinations simultaneously to overcome the randomized ensemble and guarantee attack success, provided that poor transferability properties hold. This introduces a tradeoff between the attack success and the perturbation size. The attack success increases when the attacker covers more channel-slice-architecture combinations at the cost of constraining the optimization problem further. We evaluate the effect of inference time randomness on the attack in Fig. 2.5.

2.6 EKOS Evaluation

We evaluate EKOS in two scenarios: through (1) end-to-end open-source (white-box) models (Sec. 2.7) in a simulated environment and a physical over-the-air environment, and using (2) black-box commercial VAs (Sec. 2.8). We design the evaluation in each scenario to answer these questions:

Q1: Does EKOS reduce the accidental activation instances? – Sec. 2.7, 2.8.

Q2: Does EKOS increase the cost of generating an *adaptive* adversarial activation attack? – Sec. 2.7, 2.8.

Q3: What is the performance overhead of EKOS in terms of natural accuracy and latency? – Sec. 2.7, 2.7.

2.7 Open-Source Models

We implement EKOS on open-source models and datasets.

Experimental Setup

Keyword Spotting. We use Google's Speech Commands dataset [83] for training and testing KWS models. The dataset consists of approximately 65,000 one-second long utterances of 30 short words, from thousands of different speakers. Similar to prior work, we select 12 labels: {*yes, no, up, down, left, right, on, off, stop, go, silence, unknown*} [51, 52]. We split the data into: 80% training, 10% validation, and 10% testing (3081 samples). We use Choi et al.'s implementation of the dense (DS-CNN), 1D temporal ResNet (TC-ResNet), and 2D ResNet (TC-ResNet2D) models [52], which achieve the highest accuracy with a reduced inference time.

Simulated Environment. We simulate the over-the-air channel using *Pyroomacoustics* [84] python package[3]. This package implements the image-source model [85] to calculate the acoustic reverberation and generate the room impulse response (RIR). We generate 1000 unique RIR samples where the room dimensions, speaker,

[3] `https://github.com/LCAV/pyroomacoustics`

and microphones locations are drawn uniformly at random. During audio preprocessing, we apply background noise, RIR convolution, and random shift to the speech samples to approximate real-world scenarios.

Over-the-air Environment. In the physical setup, we evaluate EKOS on a set of commodity devices with varying background noise. We deploy EKOS on six devices (D1–D6): a MacBook Pro laptop, an iPad tablet, a Dell PC with a high-quality directional microphone (Blue Snowball)[4], a Dell laptop, a Google Pixel XL phone, and a Google Pixel 2 XL phone. The devices are distributed in a lab space (14.2x7x3.8m). All devices run a lightweight webserver and are connected to the same wireless network. The PC is the main VA (server): it requests and aggregates votes from other devices (clients).

We use two Echo Dot devices as Bluetooth speakers; the first plays the keywords and the second plays background noise at half the volume. We evaluate four background scenarios: (1) noise naturally found in the lab, including a humming AC, keyboard typing, and mouse click sounds; (2) popular English songs (music & speech); (3) Google Commands dataset noise files that include doing the dishes, biking, running water, miaowing, white and pink noise; and (4) classical music[5].

Attack against a single model. We build on Qin et al.'s implementation[6] [42] for imperceptible and over-the-air robust adversarial examples on ASR (Sec. 2.3). Note that this attack is robust only on *simulated* environments. In contrast with ASR, which involves sequence-to-sequence modeling, KWS is a single word classification task. Thus, we simply apply the cross-entropy loss (instead of the CTC loss) with a regularizer for either robustness or imperceptibility.

Attack against an ensemble. Alongside attacks on individual models, we evaluate an adaptive attacker. We consider the strongest possible threat model, where an adversary has full access to the ensemble details. This adversary targets EKOS ensemble as a whole: it calculates the overall loss by summing the predicted logits (i.e., the scores assigned to each class) across the ensemble models on the input to be attacked. Then, the attack is optimized directly on this combined loss.

[4] https://www.bluemic.com/en-us/products/snowball/

[5] https://www.youtube.com/watch?v=y1dbbrfekAM

[6] https://github.com/tensorflow/cleverhans/tree/master/examples/adversarial_asr

Architecture	**BL**	$\mathfrak{l}=1$	$\mathfrak{l}=3$	$\mathfrak{l}=4$	$\mathfrak{l}=5$
DS-CNN-M	94.61	82.53 ± 1.63	84.31 ± 1	83.95 ± 0.85	84.12 ± 0.81
TC-ResNet14	96.43	91.74 ± 1.9	93.71 ± 1.34	93.43 ± 1.12	94.13 ± 0.97
TC-ResNet2D8	96.85	84.64 ± 1.74	86.27 ± 0.54	86.51 ± 0.87	86.97 ± 0.5
TC-ResNet8	96.50	92.99 ± 1.19	93.85 ± 0.74	94.57 ± 0.71	94.52 ± 0.43
Random Arch.	–	–	95.131 ± 0.81	94.606 ± 0.67	95.141 ± 0.93

Table 2.1: Accidental activation accuracy (%) (mean±std) of an ensemble (of size $\mathfrak{l}$) in a simulated environment without enabling feature slicing.

Accidental Activation Evaluation

First, we evaluate EKOS's performance against accidental activations and compare it to the baseline (single KWS model) performance. Therefore, we exclude the feature slicing component from EKOS's pipeline in this experiment.

Simulated Evaluation. We evaluate an ensemble of $\mathfrak{l}$ models, where each model experiences a unique channel (RIR). We evaluate two scenarios: (1) the same architecture is deployed on all $\mathfrak{l}$ models, and (2) each model independently selects an architecture at random with replacement. We run the evaluation 20 times to account for randomness.

Table 2.1 shows the mean and standard deviation accuracy at $\mathfrak{l} = 1$, i.e., a single device, and at an ensemble of size $\mathfrak{l} = 3, 4, 5$, versus the unrealistic baseline (BL) accuracy when the audio is directly fed to the model (digitally rather than physically). An ensemble of size 3 outperforms the single device for all architectures. There are diminishing returns for ensembles with more than three models. Random architecture selection also outperforms individual architectures. We thus confirm that an ensemble of diverse architectures and audio channels enhances the natural accuracy of any single model.

Note that Google Commands is a multi-class and balanced dataset with 12 classes. Classifying each keyword with high accuracy means fewer errors, hence, lower *accidental* (erroneous) activations. Thus, the classification accuracy on such a dataset is an indication of robustness to accidental activations.

Over-the-air Evaluation. We play the same 3081 test samples over the air and record the six devices' microphones. We feed these samples to the four KWS architectures. Fig. 2.2 shows individual devices (D1–D6) mean accuracy and

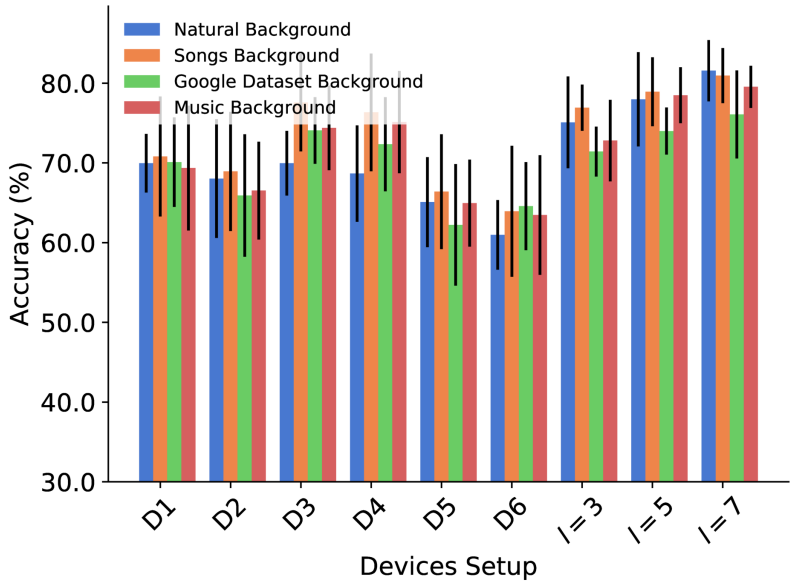


Figure 2.2: Over-the-air accuracy (mean and std) of EKOS against accidental activation for different background noises, ensemble size (l), and architecture selection. EKOS outperforms individual devices (D1–D6) under all scenarios.

standard deviation across architectures and background noise. Devices closer to the speaker (D1, D4) achieve higher accuracy than other devices (D2, D5, D6) for all noise types since they experience a higher signal-to-noise ratio (SNR). D3 also performs well since it utilizes a directional microphone.

Next, we randomly combine the six devices in an ensemble of size $l = 3, 5, 7$. Each model in l selects its architecture independently at random with replacement. We repeat the evaluation ten times to account for architecture and device selection randomness. Fig. 2.2 shows that EKOS outperforms all the individual devices under all background scenarios. Hence, the physical evaluation matches the simulated evaluation and validates EKOS' robustness against accidental activations.

Adversarial Activation Evaluation

Second, we evaluate EKOS's performance against adversarial activations: we now include the feature slicing component.

Simulated Evaluation. Before we evaluate the robustness benefits of feature slicing, we ensure that our pipeline maintains its natural accuracy. Fig. 2.3 shows

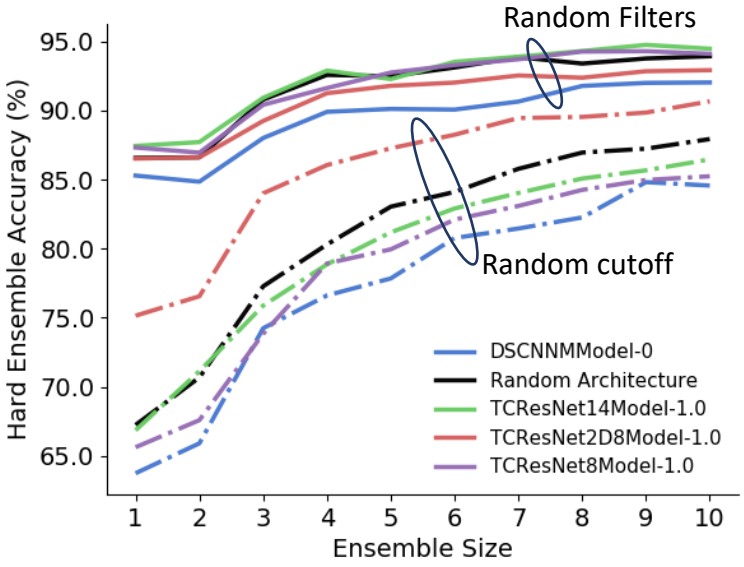


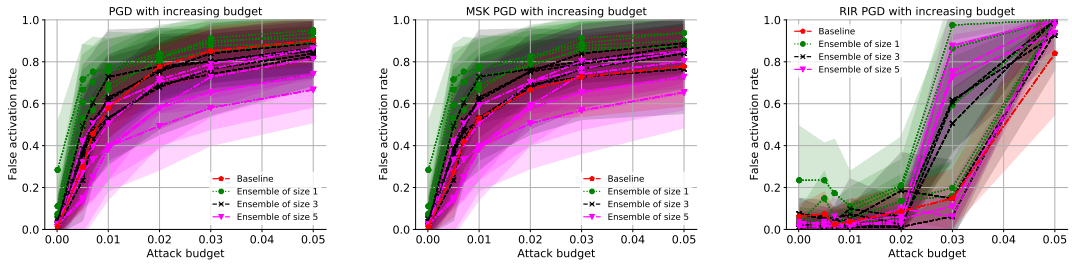
Figure 2.3: Natural mean accuracy (%) of EKOS with feature slicing (solid lines) and filter cutoff shift (dotted lines) at different architectures and ensemble sizes. The feature slices and cutoff shift are randomly selected at run-time.

the performance of EKOS at different architectures and ensemble size, with two levels of inference time randomness; (1) random filters selection from the set G, and (2) random filter cutoffs shift at run time. We apply random shifts drawn uniformly from the range $\pm$200Hz to the 4 cutoff parameters (s_l, e_l, s_h, e_h).

First, when applying random feature slicing, an ensemble of only $l = 5$ improves the individual models' accuracy by an average of 6%—corresponding to 50% error rate reduction, at all architectures. Hence, the $l = 5$ ensemble accuracy approximates the models' accuracy in Table 2.1, where no feature slicing is applied. Second, when the $\pm$200Hz random cutoff shift is applied, it deteriorates the models' accuracy. Still, the ensemble accuracy increases with the ensemble size.

Next, we evaluate EKOS against an adaptive white-box attacker. We compare the performance of Projected Gradient Descent (PGD), PGD with frequency masking, and PGD with 20 RIRs attacks. The adaptive attack is performed over an ensemble of sizes 1, 3, and 5, repeated five times for each ensemble size. All attacks use 100 iterations to accurately approximate the shortest distance to the decision boundary.

Fig. 2.4 plots the false activation rate on adversarial examples as a function of the attack budget, with standard deviation computed over different keywords. The



(a) PGD (b) PGD with masking (c) PGD with RIR EoT

Figure 2.4: False activation rate (%) of EKOS against 5 randomly selected ensembles of sizes $l = 1, 3, 5$ along-with a TCResNet8 baseline model under adversarial examples generated by: (a) PGD, (b) PGD with frequency mask, and (c) PGD with RIR attacks.

baseline shows TCResNet8 model trained on all the spectrum; we select TCResNet8 as it shows the highest robustness among the baseline architectures. The figures show that as the ensemble size increases, the adversary is unable to maintain the same attack performance compared to the baseline; i.e., the adversary needs a higher perturbation budget to reach a specific false activation rate. At higher attack budget, the perturbation power increases which leads to higher attack perceptibility. Therefore, EKOS increases the cost the adversary faces.

Most interestingly, we find that EKOS makes it hard to launch frequency masking attack efficiently (Fig. 2.4b). The mask constraint is no longer satisfied as frequency peaks are not necessarily used by individual models due to feature slicing. The PGD with frequency masking attack on EKOS is effectively relaxed to the less constrained PGD attack in Fig. 2.4a. For PGD with RIR attack, Fig. 2.4c shows a lower false activation rate, w.r.t. Fig. 2.4a and 2.4b, at the low attack budget due to RIR randomness. Note that PGD with RIR optimizes the perturbation over an EoT of the RIR transform (Eqn. 2.3); the perturbation is not guaranteed to succeed at run-time.

Fig. 7.2, in the Appendix, similarly shows the performance of the attacks in the presence of a random filter cutoff shift of $\pm$200Hz. The models exhibit behavior similar to Fig. 2.4 and cause an increased complexity for the attacker despite its relatively lower natural accuracy. Although hard to formally capture with the adaptive white-box attack evaluation, randomized filter cutoff introduces additional uncertainty for the attacker. Finally, we show in Fig. 7.1 that the attack results in an increase in the perturbation power received by individual models compared to

Attack	D1	D2	D3	D4	D5	$l = 5$
PGD	33.78	28.67	39.0	34.33	33.33	46.89
PGD_RIR	37.22	36.44	48.89	41.11	34.0	54.33

Table 2.2: Over-the-air adversarial accuracy (%) of individual device and EKOS at $l = 5$ against PGD and PGD with RIR attacks, 90 examples each.

the baseline.

Over-the-air Evaluation. We generate 90 adversarial examples from each of the PGD and PGD with RIR adaptive attacks against an ensemble of size $l = 5$. We play the adversarial examples over the air and capture the recordings from the commodity devices. We evaluate adversarial examples in a white-box setting, i.e., against the same exact models and feature filters that were used to generate them. However, the device-model assignment is done randomly. Thus, we repeat the evaluation ten times. Table 2.2 presents the average adversarial accuracy and confirms that EKOS' ensemble outperforms the individual devices against both attacks.

Next, we evaluate the attack against a randomized run of EKOS; i.e. the feature filters and KWS architectures are selected independently at random. The evaluation is repeated ten times. Fig. 2.5 shows the accuracy (mean and standard deviation) of individual devices and of EKOS's ensemble at size $l = 3, 5, 7$ for the adversarial examples and their benign samples as well. It is clear that all the devices' accuracy is higher than their values in Table 2.2 due to the randomized run.

We observe from Fig. 2.5 that EKOS's ensemble outperforms individual devices on benign and adversarial samples. Although we perform feature slicing in this experiment, accuracy on benign samples matches that of EKOS without feature slicing (Fig. 2.2), especially at $l > 3$. This is consistent with our findings from the simulated setup (Fig. 2.3); the ensemble gain compensates for the accuracy drop due to feature slicing.

Although adversarial examples are successful in the simulated setup (the model's accuracy is 0), they do not always succeed over the air (accuracy > 0 in Table 2.2 and Fig. 2.5). Moreover, while the PGD with RIR attack takes the acoustic channel into consideration, its attack success rate (1-accuracy) is not always higher than the PGD attack (without RIR). We attribute these observations to multiple

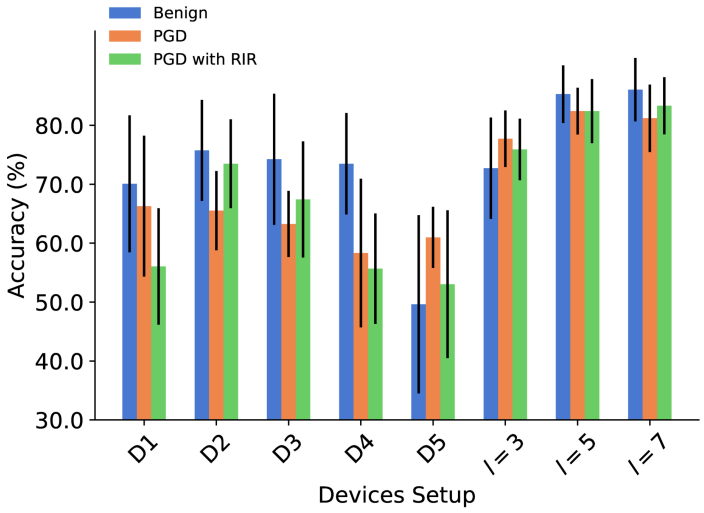


Figure 2.5: Over-the-air accuracy (mean and std) of EKOS under PGD and PGD-RIR attacks and their benign samples with random slicing filter and architecture selection.

factors: (1) the RIR simulation is only an approximation of the physical acoustic channel; (2) there are other physical transformations not taken into account, such as the microphone's non-linearity and noise; and (3) the expectation over RIR optimization does not guarantee a successful perturbation across all RIR transforms (all devices and environments). These observations match the findings from Qin et al. [42], where their adversarial examples were successful only in a simulated environment.

System Integration Analysis

Finally, we assess EKOS's deployment in terms of devices integration and end-to-end latency. EKOS latency stems from two sources: (1) model inference and (2) vote communication and aggregation. EKOS is not sensitive to device synchronization errors since it combines votes, not signals. Since the ensemble models run simultaneously and independently, the first source is dominated by the slowest device-architecture pair. The latencies of EKOS's architectures are available in prior work [52] (Table 1 and 2) and range between 1.1ms and 10.1ms.[7] We measure the

[7]The inference time is measured on a Google Pixel 1 using the TensorFlow Lite Android benchmark tool. The authors forced the model to be executed on a single core in order to emulate the always-on nature of KWS.

end-to-end latency ΔT by running an ensemble of size $l = 10$ on our set of devices; some devices run more than one model simultaneously. The setup is as follows: D1 runs three models, D4 runs two models, D5 and D6 run a single TensorFlow-Lite model each, and D3 (the main VA server) runs three models and aggregates the votes.

We performed 100 inference requests, the average latency ΔT is 0.32s$\pm$0.25s; the median, max, min are 0.21s, 1.53s, 0.20s, respectively. EKOS's latency ΔT is consistent with the latency window it takes the cloud KWS module to verify the local activation and to perform "Echo Spatial Perception[8]" to coordinate multiple Echo devices in the same environment. Hence, EKOS's latency does not degrade the user experience. Moreover, EKOS latency does not increase linearly with the number of devices; it is not accumulative. Thus, introducing more devices to EKOS will not necessarily increase its latency unless the new device forms a critical path.

2.8 Commercial Voice Assistants

In this section, we extend our evaluations of EKOS to commercial VAs and their keywords. This evaluation is challenging since we do not have access to their dataset, and there is no API access to the local KWS engine. Moreover, since the commercial models are not trained with feature slicing transformation, we cannot evaluate EKOS end-to-end; it is only feasible to assess the physical environment effect on accidental and adversarial activations. We do not apply any feature transformation or pre-processing on the evaluated keywords.

Experimental Setup

Our setup comprises 5 Echo devices: 4 Echo Dot (3^{rd} Gen), and one Echo tower (1^{st} Gen), distributed in a lab space (Fig. 2.6). The Bluetooth speaker is located in the middle of the room, and the Echo devices are located at 0.7, 3, 3, 2.7, and 2.6m away from the speaker. We choose Amazon's Echo devices because they can be activated by four different keywords: {*Alexa, Echo, Amazon, Computer*}, hence, enabling a comprehensive study. We automate the activation detection using a digital photosensitive sensor attached to the device's light rim. Once a device detects the keyword, its rim light turns on, and the sensor captures the change in

[8] `http://tiny.cc/a7trvz`

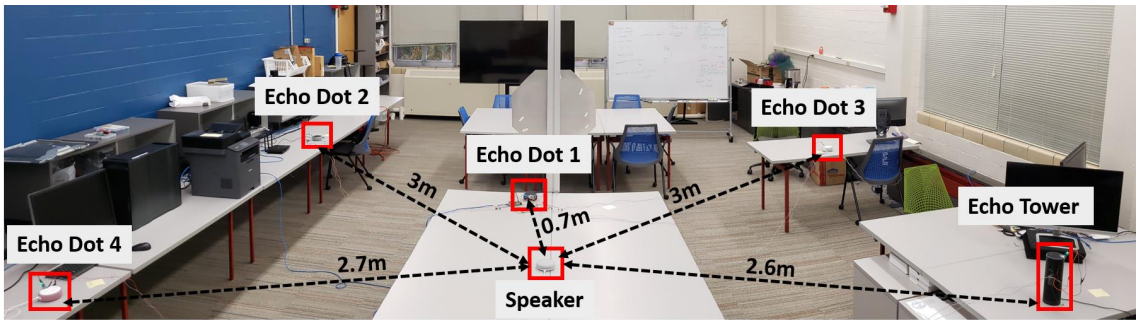


Figure 2.6: Commercial (Amazon's Echo) VAs setup of 4 Echo Dots, 1 Echo tower, and a Bluetooth speaker.

light. The setup is controlled by a Raspberry Pi 4 Model B that plays the audio sample on the Bluetooth speaker and records the Echo devices' activations via the sensors' output.

We run the experiment on the local (offline) and the local+cloud (online) KWS models. Since the VAs operation is non-deterministic [37, 48], we repeat the offline (online) experiment three (ten) times and report the average values.

Evaluation Dataset

Positive Samples. We generate two sets of positive samples for each of the four keywords, namely Positive–TTS and Positive–Speech. In the first set, Positive–TTS, we use text-to-speech APIs from Google, Amazon, and IBM to generate 77 samples for each of the four keywords in different voices (while synthetic, the samples sound natural to the ear).

We extract the second set, Positive–Speech, from conventional speech recognition datasets – Librispeech, VCTK, Common Voice, TED-LIUM, and M-AILABS. We search the transcriptions for the keywords. We use the Montreal Forced Aligner[9] to align the speech with its transcript and extract the keyword utterance. The Positive–Speech set has 104, 138, 405 samples for *Amazon*, *Echo*, and *Computer*, respectively. We found no samples for *Alexa*. Since these datasets were not manually validated, we curate them by discarding samples that do not activate any of the five devices.

[9] `https://montreal-forced-aligner.readthedocs.io/en/latest/first_steps/example.html`

Devices	Alexa				Amazon				Computer				Echo			
	Offline		Online		Offline		Online		Offline		Online		Offline		Online	
	TP	FA	TP	FA	TP	FA	TP	FA	TP	FA	TP	FA	TP	FA	TP	FA
Echo Dot1	68	44	67	20	140	75	125	89	181	326	169	217	89	75	89	49
Echo Dot2	60	31	61	14	120	48	115	39	**208**	**136**	180	86	**79**	**23**	77	36
Echo Dot3	51	27	50	31	99	29	**111**	**22**	163	130	**172**	**55**	81	36	68	34
Echo Dot4	57	22	49	3	86	6	65	4	93	71	85	54	62	30	50	11
Echo tower	59	44	66	18	92	37	95	50	189	156	167	202	72	69	70	54
Ensemble	**61**	**21**	**63**	**9**	**113**	**23**	110	22	174	128	163	75	83	31	**77**	**24**
Class Size	*Pos:* 69 – *Neg:* 985				*Pos:* 147 – *Neg:* 808				*Pos:* 263 – *Neg:* 1738				*Pos:* 105 – *Neg:* 596			

Table 2.3: Amazon Echo's offline (local) and online (cloud) KWS performance at different keywords. TP = true-positive, FA = false-activation. *Pos:* the positive class sample size, and *Neg:* the accidental activation class sample size. The TP and FA values that result in the highest accuracy per keyword are displayed in **bold**.

Negative Samples. Generally speaking, any speech utterance other than the keyword is a negative sample. However, in this experiment, we focus on worst-case samples with a high probability of incorrectly activating the device. Thus, the false activation (FA) rates we report (e.g., in Table 2.3) are higher than expected normal operation values. We extract the accidental activation dataset that Dubois et al. [48] identified in TV shows. We also use Schönherr et al.'s crafted accidental activation triggers [37]. This dataset consists of n-gram English phrases that are phonetically close to the keyword. We use text-to-speech APIs to synthesize these phrases in different voices, as done for Positive–TTS samples.

Adversarial Examples. We use the adversarial examples generated by Devil's Whisper attack [41] on Amazon Echo[10], which is an over-the-air robust attack.

Accidental Activation Analysis

Local Activation. We evaluate the local KWS model performance by disconnecting the Echo devices from the Internet. When an utterance activates the local model, the rim light turns red, and the device plays an error message. Table 2.3 shows the true-positive (TP) and false-activation (FA) counts along with their class sizes for the individual devices and their ensemble. The ensemble decision is a majority vote; i.e., it is activated when at least 3 out of 5 devices are activated.

[10] https://github.com/RiskySignal/Devil-Whisper-Attack/tree/master/AEs

Devices	Alexa				Amazon				Computer				Echo			
	Offline		Online		Offline		Online		Offline		Online		Offline		Online	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Echo Dot1	95.73	75.15	97.92	86.05	91.45	77.42	88.36	69.17	79.63	47.03	84.46	52.44	87.07	66.12	90.83	73.55
Echo Dot2	96.20	75.00	97.97	85.10	92.11	76.06	92.6	76.46	**90.44**	**68.48**	91.53	67.94	**93.06**	**76.56**	90.8	70.56
Echo Dot3	95.70	69.24	95.33	67.24	91.97	72.09	**93.97**	**79.31**	88.51	58.63	**92.70**	**70.10**	91.39	72.68	89.86	65.72
Echo Dot4	96.77	77.15	97.84	81.10	92.98	72.02	91.04	60.14	87.97	43.59	88.37	42.07	89.63	63.04	90.67	60.55
Echo tower	94.88	68.68	97.92	85.75	90.33	66.57	89.32	65.00	88.51	62.16	85.12	53.09	85.4	58.68	87.35	61.42
Ensemble	**97.19**	**80.40**	**98.57**	**89.32**	**94.03**	**79.85**	93.8	78.65	89.16	61.54	91.23	64.96	92.49	75.94	**92.57**	**74.8**
Class Size	*Pos:* 69 – *Neg:* 985				*Pos:* 147 – *Neg:* 808				*Pos:* 263 – *Neg:* 1738				*Pos:* 105 – *Neg:* 596			

Table 2.4: Amazon Echo's offline (local) and online (cloud) KWS performance at different keywords in terms of the accuracy and F1 scores (%). *Pos:* the positive class sample size, and *Neg:* the accidental activation class sample size. The highest Acc and F1 scores per keyword are displayed in **bold**.

The devices closer to the speaker such as Echo Dot1, the closest device to the speaker, have high TP values and also high FA for all keywords, and vise versa such as Echo Dot4. Hence, the user's experience and privacy are at odds with respect to the device's proximity to the user. On the other hand, the majority vote ensemble has a lower number of misactivations (FA) than most of the devices. Table 2.4 shows the accuracy and F1 scores of the individual devices and their ensemble. The ensemble accuracy and F1 are higher than the five devices at all keywords except *Computer*. For the keyword *Echo*: The ensemble accuracy is very close to Echo Dot2 and is higher than the other four devices. Hence, EKOS's ensemble achieves the two-fold objective: it protects the user's privacy with fewer FA without sacrificing the utility.

Cloud-based Activation. We reconnect the devices to the Internet to evaluate cloud KWS on both authorized and unauthorized samples. Table 2.3 shows that the numbers of FA are significantly lower than the local KWS model for almost all devices and keywords. Local activations not confirmed by the cloud model are transcribed on the voice history page with "*Audio was not intended for Alexa*." Although the cloud KWS verification enhances the user experience by limiting unwanted and unexpected interactions with the user, it does not necessarily mitigate the privacy concerns: private conversations are still being sent to the cloud.

To quantify the privacy leakage, we analyze the misactivations duration, i.e., the time during which the rim light stays on. Fig. 2.7 shows the duration distribution per keyword. We find that the duration is concentrated from 1.6s to 10s with 6.33s median and 5.75s mean, with some samples reaching up to 86s. Hence, the

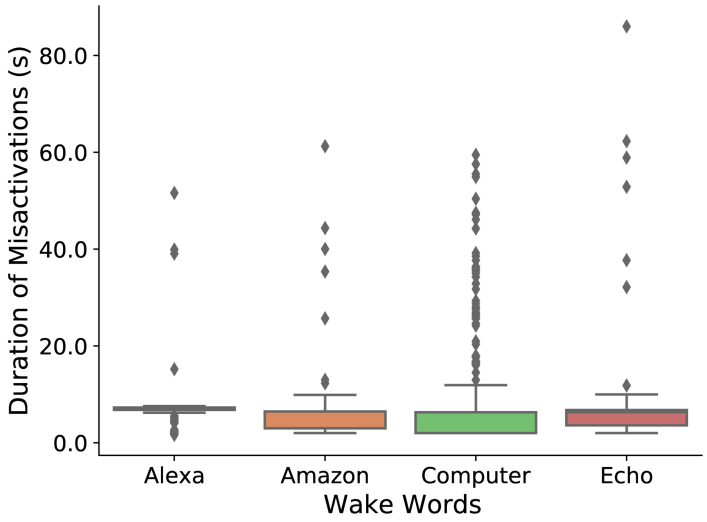


Figure 2.7: Duration (seconds) of Cloud-based Misactivations of Echo VAs per Keyword.

misactivations are long enough to leak private conversations. The FA rate and the misactivation duration quantify the magnitude of privacy leakage.

We believe that our experiment could capture some worst-case misactivation durations, as compared to prior research [48]; the devices are distributed in a large room with uncontrolled background noise, unlike prior work [48] that places the VAs and the speaker in a small isolated cabinet.

The cloud TP values, however, are in the same range as the local KWS model. This observation is unsurprising since the positive activation is mainly controlled by the local model; the cloud model only confirms or discards the local activation. In other words, the false-negative (misdetection) rate, ($\mathsf{FNR} = 1 - \mathsf{TPR}$), cannot be improved by the cloud model, which explains why the local KWS model's design favors TPR rather than FPR. We attribute the small differences in the TP values between local and cloud models to the non-deterministic behavior of VAs and the uncontrolled environmental variations in the lab, such as background noise.

Finally, similar to the local model, EKOS' ensemble outperforms the individual devices; it has a lower FA and a higher TP. Table 2.4 also shows that the ensemble accuracy and F1 scores are superior to most of the devices. Hence, the KWS ensemble is a practical solution that can enhance commercial VAs performance and preserve the user's privacy.

Adversarial Examples on Commercial VAs

We play Devil's Whisper 2 adversarial examples on the keyword *Echo*, which reports a 0.5 adversarial success rate (SR) in their original setup. Their adversarial SR on our setup (Fig. 2.6) is 0.7, 0, 0, 0, and 0.1 on the individual devices, and SR$=0$ on the ensemble. We relocate the devices such that they are at a 1m distance from the speaker; the SR increases to 0.6, 0.7, 0.15, 0, and 0.9, respectively, and the ensemble SR$=0.45$. Hence, the ensemble vote could reduce the adversarial activations as well, even without feature slicing and architecture diversity, in a non-adaptive attack setting.

2.9 Discussion

In the following, we discuss the tradeoffs in deploying EKOS as well as some limitations and future work directions.

Deployment Analysis

EKOS is a flexible system; it enables a custom-tailored operation with tunable utility, usability, and privacy trade-off.

Number of devices N**:** Ensemble models can be either centralized in the main VA or distributed over a set of smart devices in the environment. The former is a compact and, possibly, more usable setting. On the other hand, the latter minimizes the VA footprint via distributed computation and provides more spatial and hardware diversity. Thus, N is a control parameter the user can tune to optimize the robustness, utility, and computational cost. We evaluate EKOS on closely located microphones on a single VA in Appendix 7.1.

Feature Slicing: The user can enable (disable) the feature slicing filters to include (exclude) the adversarial activation threat model. However, as shown in Fig. 2.3, 2.5, with $\mathsf{l} \geqslant 5$, the ensemble with feature slicing enabled reaches the baseline natural accuracy without feature slicing (Table 2.1, Fig. 2.2). Also, the user can enable (disable) the random filter cutoffs shift for a more robust (accurate) operation – refer to Fig. 2.3.

Number of architectures K**:** As shown in Table 2.1 and Fig. 2.2, architecture diversity enhances the ensemble accuracy and robustness. Yet, channel diversity

and feature slicing also contribute to EKOS' accuracy and robustness. Hence, the number of stored architectures can be set to only one without sacrificing EKOS's performance when device storage is limited.

Number of models l: As shown in Table 2.1 and Fig. 2.2, 2.3, the optimal number of models ranges from 3 to 5. Hence, EKOS is a practical system, it enhances the VA robustness at an overhead of $l \leqslant 5$ models deployed on $N \leqslant l$ devices.

Computation Overhead: EKOS adds computation overhead to the commodity devices. However, the client devices run the KWS model only when they receive a request from the main VA (Sec. 2.5). Hence, the computation overhead will be limited to infrequent true-positive and false-positive incidents. Moreover, the centralized mode of operation only loads the main VA, which is a plug-in device.

Limitations and Future Research

Distance Bounding. A device far from the user may increase the false-negative rate due to its low SNR. Hence, the ensemble devices should perform a periodic distance bounding check. Although we do not include this component in this chapter, distance bounding is a well-studied problem with existing engineering and cryptographic solutions [86, 87].

Other Robustness Measures. EKOS can be integrated with other measures to increase robustness such as adversarial training and data augmentation. Another interesting integration is to perform voice recognition where the VA only accepts commands that match the authorized user voiceprint. Voice recognition filters out accidental and adversarial activations from unauthorized speakers, TV, YouTube videos, etc. An adversary would need to attack both systems jointly to initiate a successful mis-activation . Hence, voice recognition will increase the attack cost and enhance the robustness.

Other Attack Models. EKOS does not directly address attacks based on the microphone non-linearity such as light commands [76] and ultrasound signals [75, 74], yet, it increases their cost. For example, the light attack will require at least $l/2 + 1$ laser beams to target the majority of the devices simultaneously. Likewise, the ultrasound attack needs to be performed at close proximity to the microphones

with speaker-microphone aligned orientation. Hence, the distributed ensemble will increase this attack cost as well.

Replay and spoofing attacks can fool the KWS using recorded or synthetic speech commands that contain the correct keyword. To address them, EKOS can incorporate a voice-liveness detector [45, 88] to distinguish human and machine-generated commands. We envision a hierarchical detection algorithm that rejects any non-human command.

2.10 Related Work

Privacy Measures for KWS. Cloud operators provide some privacy measures to minimize accidental activations. For example, VA providers give their users access to their voice commands history, where they can listen to and delete their past commands [89]. After the public backlash on manual transcription of voice commands, Google, Apple, Amazon, and Facebook suspended the default enrollment and are currently giving the users opt-in and opt-out choices [90]. Recently, Google has enabled a tunable keyword sensitivity setting to give the users control over the utility and privacy tradeoff [91]. Yet, these measures do not address the privacy threats of VA misactivations and do not meet the users' expectations of hands-free interactions. Recently, Apple has announced [92] that Siri will process speech locally on the user's device to mitigate these privacy concerns.

Researchers have also proposed external privacy control systems to reduce misactivation incidents. Karmann proposes a small add-on device that jams the VA's microphones and lifts up the jamming when it detects a user-customized keyword [93]. However, it suffers from the same privacy threats of KWS misactivation. Wu et al. [94] propose an audio-visual speech recognition by analyzing lip movement in a sensor-fusion algorithm. Similarly, Mhaidli et al. [95] use the gaze direction and voice volume level as interpersonal communication cues of the user's intent to activate the VA. These efforts, however, introduce another privacy threat by adding an always-on camera in the user's private environment. Coucke et al. propose Snips, a private-by-design system [96] that processes the commands locally without cloud interactions. Yet, it cannot be integrated into the commercial VAs.

Adversarial Example Defenses. Generic defenses like adversarial training [44] and verifiable robustness [97, 98, 99] are largely detached from the real world; they assume the adversarial capabilities to respect an ℓ_p-norm ball constraint. Yet, attacks in the physical space can map to large ℓ_p-norm perturbations in the digital space. Moreover, these approaches are hard to train, and come with significant performance loss limiting their adoption. Others have explored defenses specific to the audio domain. Bhattacharya et al. [77] propose a stochastic compression technique. Liveness detection distinguishes commands coming from a human or an audio speaker, either via spectrum analysis [88, 45] or motion sensing [100], or detecting the audio speaker magnetic field [101]. This technique introduces additional privacy threats of sensing human activities and is ineffective against accidental activations.

2.11 Conclusion

We took two complementary approaches to tackle two forms of misactivations: accidental and adversarial. Both approaches rely on a diversity of sensors and models used to perceive the environments. However, they differ in that adversarial activations require provable guarantees of increased costs, which we achieve by exploiting a physical property of the audio wave (replicas of speech at different harmonics) rather than taking an ML approach. We hope this chapter paves the way for a new class of approaches that systematically characterize environmental constraints to improve ML robustness. In the following chapter, our focus shifts to examining the privacy implications inherent in cloud-based *speech transcription*. Unlike voice activation, speech transcription entails the transformation of continuous and extended speech recordings, such as those from meetings or interviews, into textual transcripts.

Chapter 3

Pr$\epsilon\epsilon$ch: Privacy-Preserving Speech Transcription

3.1 Introduction

In this chapter, we broaden our investigation of the privacy risks associated with voice-enabled technologies to include cloud-based speech transcription. Our focus extends from individual device interactions to the broader implications of outsourcing speech data processing to the cloud. Investigating privacy violations within the context of speech transcriptions enhances our understanding of the multifaceted nature of privacy concerns in machine learning applications, particularly in the context of speech and voice applications.

New advances in machine learning and the abundance of speech data have made Automated Speech Recognition (ASR) systems practical and reliable [102, 6]. ASR systems have achieved a near-human performance on standard datasets [102, 6], at a scale. This scalability is desirable in many domains, such as journalism [103], law, business, education, and health care, where cost, delay, and third-party legal implications [9] prohibit the application of manual transcription services [10]. For example, recent research has identified private voice transcription as one of the challenges journalists face when interviewing sensitive sources [103].

Several companies, such as Google and Amazon, provide online APIs for speech transcription. This convenience, however, comes at the cost of privacy. A speech recording contains acoustic features that can reveal sensitive information about the user, such as age, gender [104], emotion [105, 106], accent, and health condi-

tions [8]. The acoustic features are also biometric identifiers of the speakers [107], enabling speaker identification and impersonation [7]. Additionally, the textual content of speech can be sensitive [9]. For example, medical recordings can contain private health information about patients [10], and business recordings can include proprietary information. Current cloud services already support several speech processing APIs like speaker identification and diarization. They also support text analysis APIs, such as topic modeling, document categorization, sentiment analysis, and entity detection (Sec. 3.3), that can extract sensitive information from text. Applying these APIs to the recorded speech can significantly undermine the user's privacy.

Offline and open-source transcription services, like Deep Speech [108], solve these privacy challenges as the speech files never leave the user's trust boundary. However, we find that their performance does not match that of a cloud service provider [109], especially on real-world conversations and different accents (Sec. 3.2). Thus, the primary goal of this chapter is to: *provide an intermediate solution along the utility-privacy spectrum that uses cloud services while providing a formal privacy guarantee.*

This chapter presents Preech (Privacy-Preserving Speech) as a means to achieve this goal; it is an end-to-end speech transcription system that: (1) protects the users' privacy along the acoustic and textual dimensions; (2) improves the transcription performance relative to offline ASR; and (3) provides the user with control knobs to customize the trade-offs between utility, usability, and privacy.

Acoustic Privacy: Preech applies voice conversion to protect the acoustic features of the input speech file and ensure noise indistinguishability.

Textual Privacy: Preech segments and shuffles the input speech file to break the context of the text, effectively transforming it into a bag-of-words. Then, it injects dummy (noise) segments to provide the formal privacy guarantee of differential privacy (DP) [110].

We evaluate Preech over a set of real-world datasets covering diverse demographics. Our evaluation shows that Preech provides a superior transcription accuracy relative to Deep Speech, the state-of-the-art offline ASR. Also, Preech prevents cloud services from extracting any user-specific acoustic features from the speech. Finally, applying Preech thwarts the learning of any statistical models or sensitive information extraction from the text via natural language processing tools.

The main contributions of this chapter are:
(1) End-to-end practical system: We propose Preech, a new end-to-end system that provides privacy-preserving speech transcription at an improved performance relative to offline transcription. Specifically, Preech shows a relative improvement of 2% to 32.52% (mean 17.34%) in word error rate (WER) on real-world evaluation datasets over Deep Speech, while fully obfuscating the speakers' voice biometrics and allowing only a DP view of the textual content.
(2) Non-standard use of differential privacy: Preech uses DP in a *non-standard way*, giving rise to a set of new challenges. Specifically, the challenges are (1) "noise" corresponds to concrete words, and need to be added in the speech domain (2) "noise" has to be indistinguishable from the original speech (details in Sec. 3.4).
(3) Customizable Design: Preech provides several *control knobs* for users to customize the functionality based on their desired levels of utility, usability, and privacy (Sec. 3.7). For example, in a relaxed privacy setting, Preech's relative improvement in WER ranges from 44% to 80% over Deep Speech (Sec. 3.7).

3.2 Speech Transcription Services

We first provide some background on online and offline speech transcription services. Next, we present a utility evaluation using standard and real-world speech datasets.

Background

Speech transcription refers to the process of extracting text from a speech file. ASR systems are available to the users either through cloud-based online APIs or offline software.
(1) Cloud-Based Transcription: We utilize two cloud-based speech transcription services: Google's Cloud Speech-to-Text and Amazon Transcribe.
(2) Offline Transcription: We consider the Deep Speech architecture from Baidu [108], which is trained using Mozilla's [1] Common Voice dataset as a representative offline transcription service. This dataset is crowdsourced and open-source. Specifically, we use the Deep Speech 0.4.1 model [2] (released in January 2019). Note that we

[1] https://voice.mozilla.org/en/datasets
[2] https://github.com/mozilla/DeepSpeech

do not consider offline transcribers that are not open for general use. For example, Google's on-device speech recognizer [111] is an offline transcriber that is currently only supported on Google's Pixel devices and does not allow an API or open-source access, limiting its usability.

Notations: Let S denote the input speech file associated with a ground truth transcript T_S^g. The user can either use a cloud service provider (CSP) or an offline service provider (OSP) to obtain the transcript (denoted by T_S^{CSP} or T_S^{OSP}, respectively).

Transcription Accuracy: The standard metric for quantifying the accuracy loss from transcription is the word error rate (WER) [108]. WER treats the transcript as a sequence of words. It models the difference between the two sequences by counting the number of deleted words (D), the number of substituted words (U), and the number of injected words (I). If the number of words in T_S^g is W, WER is given as: $\frac{D+U+I}{W}$.

Utility Comparison

In this section, we empirically evaluate the utility gap between the CSP and the OSP over a wide range of standard and real-world datasets. We use these datasets throughout the chapter.

Standard Datasets: These datasets include (1) the TIMIT-TEST subset [112], (2) a subset from Librispeech *dev-clean* dataset [113], and (3) the DAPS dataset [114]. TIMIT-TEST [3] subset comprises of 1344 utterances by 183 speakers from eight major dialect regions of the United States. The LibriSpeech subset consists of eleven speakers, 20 utterances each. For DAPS, we use the evaluation subset prepared for the 2018 voice conversion challenge [115] that consists of five scripts read by ten speakers: five males and five females.

Real-world Datasets: We also assess the real-world performance of both transcription services on non-American accent datasets and real conversations among speakers of different demographics. For the accented datasets, we evaluate 200

[3] `https://catalog.ldc.upenn.edu/LDC93S1`

	Datasets	Google	AWS	Deep Speech
Standard	LibriSpeech	9.14	8.83	9.37
	DAPS	6.70	7.53	10.65
	TIMIT TEST	6.27	7.11	20.08
Non-Standard	VCTK p266	5.15	10.09	26.72
	VCTK p262	4.53	7.87	15.97
	Facebook 1	5.76	7.45	24.72
	Facebook 2	3.07	8.19	26.61
	Facebook 3	8.32	9.42	30.72
	Carpenter 1	9.44	9.44	25.85
	Carpenter 2	9.22	11.53	39.71

Table 3.1: WER (%) comparison of cloud services, Google and AWS, versus the state-of-the-art offline system, Deep Speech.

utterances of two speakers from the VCTK dataset [116]: speaker p262 of a Scottish accent and speaker p266 of an Irish accent. For the real-world datasets, we evaluate 20 minutes of speech from the "Facebook, Social Media Privacy, and the Use and Abuse of Data" hearing before the U.S. Senate [4]. We construct the 20 minutes by selecting three continuous chunks of speech from the hearing such that they include nine speakers: 8 senators and Mark Zuckerberg. Another real-world dataset is the Supreme Court of the United States case "Carpenter v. United States" [5]. For this dataset, we evaluate a total of 40 minutes of speech from the advocates in the case.

Accuracy Comparison: Table 3.1 presents the WER comparison results. The results show that the CSPs are superior to the OSP on all the datasets. The performance gap, however, is more significant on the non-standard datasets; the CSP outperforms Deep Speech by 60% to 80% in WER.

3.3 Privacy Threat Analysis

We study the privacy threats that a cloud-based transcription service poses while processing private speech data.

[4] https://www.commerce.senate.gov/2018/4/facebook-social-media-privacy-and-the-use-and-abuse-of

[5] https://www.oyez.org/cases/2017/16-402

Voice Analysis

The biometric information embedded in S can leak sensitive information about the speakers, including their emotional status [105, 106], health condition [8], sex [104], and even identity [107]. Furthermore, extracting this information enables critical attacks like voice cloning and impersonation attacks [117, 118]. In this section, we showcase a few representative examples of how cloud-based APIs can pose serious privacy threats to the acoustic features within S.

Speaker Diarization: CSPs utilize advanced diarization capabilities to cluster the speakers within a speech file, even if they have not been observed before. The basic idea is to (1) segment the speech file into segments of voice activity, and (2) extract a speaker-specific embedding from each segment, such that (3) segments with close enough embeddings should belong to the same speaker. We verified the strength of the diarization threat over three multi-speaker datasets: VCTK (mixing p266 and p262), Facebook, and Carpenter. We measure the performance of the IBM diarization service using Watson's Speech-to-Text API [6] via Diarization Error Rate (DER). DER estimates the fraction of time the speech file segments are not attributed to the correct speaker cluster. The DER values are 0%, 4.85%, and 1.32% for the three datasets, respectively. Hence, the API can correctly distinguish between, and cluster, the different speakers, more than 95% of the entire dataset duration despite lacking any prior information about the individual speakers.

Speaker Identification: A speaker identification task maps the speech segments in a speech file to an individual. We use the Azure Identification API, which consists of two stages: (1) user enrollment and (2) identification (whether a given voice sample matches any of the enrolled users). The enrollment stage requires only 30 seconds of speech from each user to extract their voice-print. We enrolled 22 speakers as follows: 10 from DAPS, two from VCTK, two from Carpenter, and eight from Facebook. The identification accuracy was nearly 100% for all speakers.

Speaker Cloning and Impersonation: Lastly, we applied an implementation [7] of a Tacotron-based speech synthesizer from Google [7]; a network that can synthesize

[6] `https://www.ibm.com/cloud/watson-speech-to-text`

[7] `https://github.com/CorentinJ/Real-Time-Voice-Cloning`

speech in the voice of any speaker. The network generates a target speaker's embedding, which it uses to synthesize speech on a given piece of text. In our setting, we used the network to generate the embeddings of the speakers in our evaluation datasets. Then, we synthesized eight speech utterances using the embeddings of each speaker. We enrolled the speakers in Azure's Speech Identification API using their natural voice samples and tested whether the API will map the synthesized segments to the corresponding speaker. Except for the second speaker in Carpenter, the cloned samples were successfully identified as the true speakers. Hence, the synthesizer could successfully impersonate the speakers in the eyes of Azure's API.

Text Analysis

CSPs possess natural language processing (NLP) capabilities that enable automated statistical analyses on large sets of documents. Those analyses fall into two broad categories. The first type involves identifying specific words from the transcript that correspond to sensitive information such as an address, name, and SSN using named-entity extraction [119]. The other type of analysis involves statistically analyzing the entire transcript on the whole to extract some semantic or user-identifying information. This analysis uses two types of information: the set of words (i.e., bag-of-words representation of the transcript) and their order of appearance (to capture the context).

Bag-of-Words Analysis: One of the most commonplace analysis that treats a document as a bag-of-words is *topic modeling* [120, 121]. Topic modeling is an unsupervised machine learning technique that identifies clusters of words that best characterize a set of documents. Another popular technique is *stylometry analysis*, which aims at attributing authorship (in our case, the speaker) of a document based on its literary style. It is based on computing a set of stylistic features like mean word length, words histogram, special character count, and punctuation count from the disputed document [122].

Context-based Analysis: An example of context-based analysis is sentiment analysis (understanding the overall attitude in a block of text). Text categoriza-

tion is another example; it refers to classifying a document according to a set of predetermined labels.

3.4 Preech

Our discussion in the previous sections highlights a trade-off between privacy and utility. The OSP provides perfect privacy at the cost of higher error rates, especially for non-standard speech datasets. On the other hand, clear privacy violations accompany revealing the speech recording to the CSP. Motivated by this trade-off, we present Preech, a practical system that lies at an intermediate point along the utility-privacy spectrum of speech transcription.

System and Threat Models

We consider the scenario where users have audio recordings of private conversations that require high transcription accuracy. For example, a journalist with recordings of confidential interviews is a paradigmatic user for Preech. Other examples include a therapist with recordings of patient therapy sessions or a course instructor with oral examination records of students. Preech, however, does not target real-time transcription applications. For example, voice assistants and online transcription (e.g. a live-streaming press conference) are *out-of-scope*. Thus, for our target use cases, the latency of transcription is not a critical concern.

The adversary is the CSP or any other entity having direct or indirect access to the stored speech at the CSP servers. This adversary is capable of the aforementioned voice- and text-based analysis.

Preech Overview

Preech provides an end-to-end tunable system which aims at satisfying the following design goals:

1. protect the users' privacy along the acoustic and textual dimensions;
2. improve on the transcription accuracy compared to offline models; and
3. provide the users with control knobs to customize Preech's functionality according to their desired level of utility, usability, and privacy.

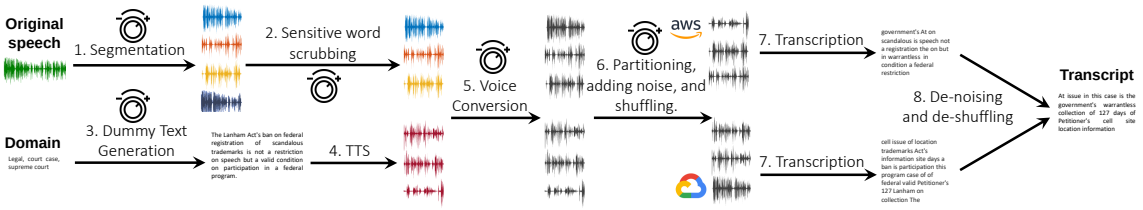


Figure 3.1: High-level overview of Preech, showing the knobs where a user can tune the associated trade-offs.

To this end, Preech applies a series of *privacy-preserving operations* to the input speech file before sending it to the CSP. Fig. 3.1 shows the high-level overview of Preech. Below, we briefly describe Preech's privacy-preserving operations.

Preserving Textual Privacy

Preech protects the privacy of the textual content of an input speech file S through the following three operations:

Segmentation and shuffling: Preech breaks S into a sequence of segments, denoted by $\mathbb{S}$. This is followed by shuffling the segments to remove all ordering information. Thus, segmenting and shuffling S transform its textual content into a bag-of-words representation.

Sensitive word scrubbing (SWS): First, Preech applies the OSP to identify the list of sensitive keywords that contain numbers, proper nouns, or any other user-specified words. Next, Preech applies keyword spotting, KWS, (identify portions of the speech that correspond to a keyword) to each of the segments in $\mathbb{S}$. Only the segments that do not contain a keyword pass to the CSP for transcription.

Dummy word injection to ensure differential privacy: The bag-of-words representation of a transcript corresponds to its word histogram (Sec. 3.4). As discussed in Sec. 3.3, several statistical analyses can be built on the word histogram of the transcript T_S^{CSP} such as topic modeling or stylometry analysis. Thus, protecting the privacy of this word histogram is a primary focus of Preech, and the privacy guarantee we choose is that of differential privacy. To this end, Preech ensures DP by adding a suitable amount of dummy words to S before sending it to the CSP. This way, the CSP is allowed only a differentially private view of the word

histogram and any subsequent statistical model built over it (by Thm. 3.4.1 in Sec. 3.4).

The main challenge in this setting is that the dummy words must be added in the speech domain, which Preech addresses as follows. First, Preech estimates the general domain of the text for S (specifically its *vocabulary*, details in Sec. 3.4) from T_S^{OSP}. Next, it generates dummy text segments using a state-of-the-art NLP language model. Finally, Preech applies text-to-speech (TTS) transforms to these dummy segments and adds them to S. However, leaving it just at this would be insufficient as the CSP can potentially distinguish between the two different sources of speech (TTS generated dummy segments and segments in S) based on their acoustic features. Therefore, Preech provides the user with multiple options to synthesize *indistinguishable* dummy segments, namely (1) voice cloning [7], and (2) voice conversion [123, 124]. These options offer different trade-offs between utility, usability, and privacy (Sec. 3.4 and 3.4). As stated in Sec. 3.3, text-based attacks exploit individual sensitive words or the order of the words or the word histogram. Thus, from the above discussion, Preech protects privacy along all three dimensions (evaluation results in Sec. 3.7).

Preserving Voice Privacy

Voice conversion, VC, is a standard speech processing technique that transforms the voice of a source speaker of a speech utterance to that of another speaker. Preech applies voice conversion to fulfill a two-fold agenda. First, it obfuscates the sensitive voice biometric features in S. Second, VC ensures that the dummy segments (noise added to ensure differential privacy) are acoustically indistinguishable from the original speech file segments. There are two main categories in voice conversion: one-to-one VC, and many-to-one VC (Sec. 3.4).

End-to-End System Description

Fig. 3.1 depicts the workflow of Preech. Given a speech file S, the first step **(1)** is to break S into a sequence of disjoint and short speech segments, $\mathbb{S}$. This is followed by **(2)** sensitive word scrubbing where speech segments containing numbers, proper nouns, and user-specified keywords are removed from $\mathbb{S}$. Next, **(3)** given the domain of S's textual content (its vocabulary), Preech generates a set of text segments (as is suitable for satisfying the DP guarantee as discussed in Sec. 3.4),

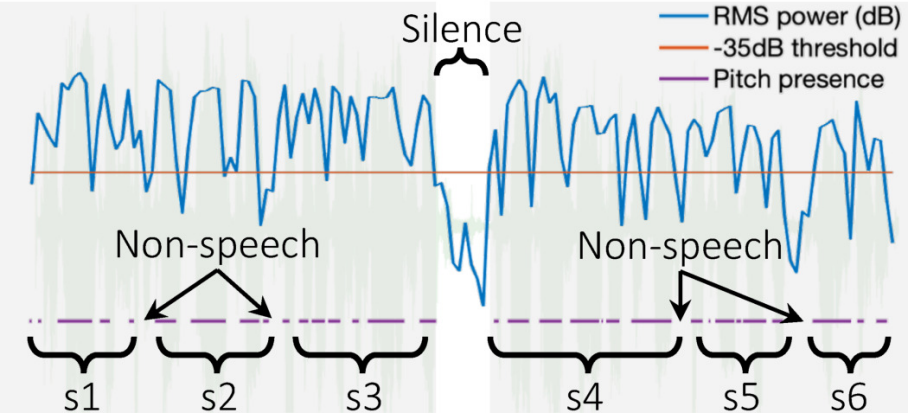


Figure 3.2: An illustration of Preech's segmentation algorithm. The coarse segments in light gray. The absence of pitch information indicate non-speech instances, which further breaks down the coarse segments into finer segments.

and subjects it to TTS transformation (**4**). At this point, Preech has audio segments for the input speech, $\mathbb{S}$, as well as the dummy segments, $\mathbb{S}_d$. If the user also wants to hide the voice biometric information in S, Preech applies (**5**) voice conversion over all the segments in $\mathbb{S} \bigcup \mathbb{S}_d$ to convert them to the same target speaker. This process hides the acoustic features of S and ensures that the segments in $\mathbb{S}$ and $\mathbb{S}_d$ are indistinguishable. This is followed by Preech partitioning $\mathbb{S}$ across $N > 0$ non-colluding CSPs (Sec. 3.4). This partitioning reduces the number of dummy segments that are required to achieve the DP guarantee (Sec. 3.4). Next, Preech adds a suitable amount of dummy segments from $\mathbb{S}_d$ to each partition $\mathbb{S}_i, i \in [N]$ and shuffles them. Additionally, Preech keeps track of time-stamps TS_i of the dummy segments and order of shuffling $Order_i$ for each such partition (**6**). After obtaining the transcript (**7**) for each partition from the N CSPs, Preech removes $\mathbb{S}_d$'s transcripts and de-shuffles the remaining portion of the transcript using TS_i and $Order_i$, and outputs the final transcript to the user (**8**).

In what follows, we elaborate on the key components of Preech, namely segmentation, sensitive word scrubbing, DP word histogram release, and voice conversion.

Segmentation Algorithm

A key component of Preech is breaking the textual context by segmenting S. We represent S as a sequence of segments $\mathbb{S}$, where each segment can contain multiple words. Preech applies a hierarchical segmentation approach that starts with a stage

of silence detection based on the energy level, followed by pitch detection to detect speech activity for finer segmentation. The mechanism is illustrated in Fig. 3.2.

We define a *period of silence* as the time duration when the RMS power of the speech signal drops below -35 dB for at least 500ms. The initial segmentation stage detects such silence periods from S resulting in coarse segments. A human speech signal can be viewed as a modulated periodic signal where the signal period is referred to as the *glottal cycle* [125]. In the second stage, Preech uses the existence of glottal cycles [126] to detect human voice, which breaks down the coarse segments into finer ones. A time duration of at least 20 ms without the presence of glottal cycles is regarded as *non-speech*.

As some segments might be abrupt or too short to allow for correct speech recognition, Preech performs two additional optimization steps. First, it merges nearby fine segments to ensure a minimum length per segment. Second, it does not partition segments at the boundaries of the identified human speech and allows 40 ms of non-speech to be included at the beginning and the end of each segment. The formal algorithm is outlined in Algorithm 1.

Control Knob: Segmenting S presents with a trade-off – smaller segments result in better privacy guarantee at the expense of deteriorated transcription accuracy due to semantic context loss. Preech allows the user to tune the *minimum length of the segments as a means to control this trade-off.*

Sensitive Word Scrubbing

Preech performs sensitive word scrubbing (SWS) as follows. First, it obtains the offline transcript of S, T_S^{OSP}. Next, it applies named entity recognition (NER) on T_S^{OSP}. NER is an NLP technique that seeks to locate and classify named entities in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, monetary values, etc. Preech also gives the option for users to specify some keywords of their choice. This allows customization of the sensitive keyword list as users have subjective ideas of what they might consider sensitive.

After the list of sensitive words is finalized, Preech applies keyword spotting (KWS) on the segments. KWS is needed for the following three reasons. First, KWS is used to spot the user-defined keywords which cannot be identified by

Algorithm 1 Hierarchical Speech Segmentation

Input: Speech file S
Minimum segment duration $l_{\mathbb{S}}$
Output: Sequence of speech segments $\mathbb{S}$
Stage 1: Silence Detection
1: Identify the timestamps $t_S = \{t_S^i\}$ corresponding to periods of silence in S
2: Divide S into a sequence of coarse segments $\mathbb{S}_1$ such that each segment is bounded by sequential timestamps from t_S
Stage 2: Pitch Detection
3: $\mathbb{S} = \varnothing$
4: **for** segment $\mathbb{S}_1^i \in \mathbb{S}_1$
5: Divide each $\mathbb{S}_1^i$ into a sequence of finer segments, $\mathbb{S}_2^i$, by identifying glottal cycles with buffers of *non-speech* at the segment boundaries
6: **do**
7: merge adjacent segments in $\mathbb{S}_2^i$ into a longer segment $\mathbb{S}^i$
8: **while** $(length(\mathbb{S}^i) < l_{\mathbb{S}})$
9: $\mathbb{S} = \mathbb{S} + \mathbb{S}^i$ ▷ "+" denotes that the segments are added in a sequence ordered by their timestamps
10: **end for**
11: Return $\mathbb{S}$

NER. Second, the initial T_S^{OSP} is generated on S without segmentation to achieve the highest estimation accuracy. However, for Preech, we need to identify the segments containing the keywords. Finally, the OSP might not transcribe the named-entities correctly at all locations. For example, the name "Carpenter" might be repeated 20 times in S, while the OSP transcribes it accurately only five times. KWS has higher accuracy in spotting keywords than the OSP's transcription accuracy.

Control Knob: KWS takes the list of keywords and matches them phonetically to a speech file based on a sensitivity score. This sensitivity score sets a threshold for the phonetic similarity required for a keyword to be spotted. A low score results in false positives by flagging phonetically similar words as keywords which degrades the utility by transcribing non-sensitive segments using the OSP. Conversely, a high score could result in some keywords being missed and revealed to the CSP. Hence, the sensitivity score is a trade-off parameter between privacy and utility (Sec. 3.7).

Differentially Private Word Histogram

We define $\mathit{vocabulary}$, $\mathcal{V}$, to be the domain of non-stop and stemmed words from which T_S^g is constructed. Let c_i denote the frequency of the word $w_i \in \mathcal{V}$ in T_S^g. As is typical in the NLP literature, we model the transcription as a bag of words: $BoW = \{w_i : c_i | w_i \in \mathcal{V}\}$. Additionally, let H represent $[c_i]$—the count vector of BoW. In other words, the bag of words model represents a histogram on the vocabulary, i.e., a mapping from $\mathcal{V}$ to $\mathbb{N}^{|\mathcal{V}|}$.

Privacy Definition

As discussed in Sec. 3.3, the aforementioned word histogram is sensitive and can only be released to the CSP in a privacy-preserving manner. Our privacy guarantee of choice is DP which is the de-facto standard for achieving data privacy [110, 127, 128]. DP provides provable privacy guarantees and is typically achieved by adding noise to the sensitive data.

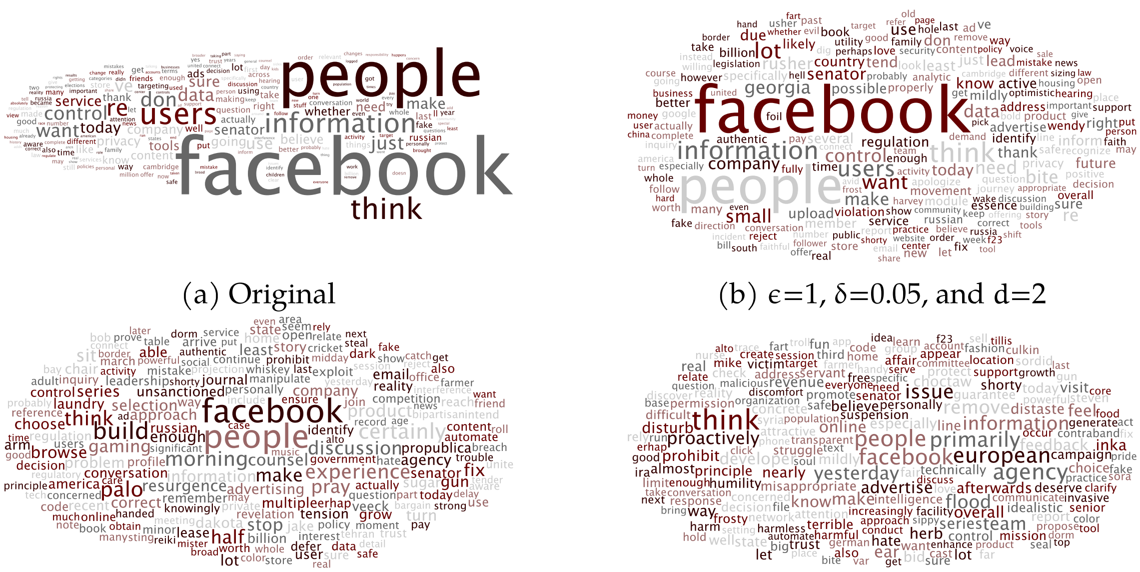
Definition 3.4.1 ((ϵ, δ)-differentially private d-distant histogram release)**.** A randomized mechanism $\mathcal{A} : \mathbb{N}^{|\mathcal{V}|} \to \mathbb{N}^{|\mathcal{V}|}$, which maps the original histogram into a noisy one, satisfies (ϵ, δ)-DP if for any pair of histograms H_1 and H_2 such that $\|H_1 - H_2\|_1 = d$ and any set $O \subseteq \mathbb{N}^{|\mathcal{V}|}$,

$$Pr[\mathcal{A}(H_1) \in O] \leqslant e^{\epsilon} \cdot Pr[\mathcal{A}(H_2) \in O] + \delta. \tag{3.1}$$

In our context, the DP guarantee informally means that from the CSP's perspective, the observed noisy histogram, $\tilde{H}$, could have been generated from any histogram within a distance d from the original histogram, H. We define the set of all such histograms to be the ϵ-indistinguishability neighborhood for H. In other words, from $\tilde{H}$ the CSP will not be able to distinguish between T_S^{CSP} and any other transcript that differs from T_S^{CSP} in d words from $\mathcal{V}$.

An important result for differential privacy is that any post-processing computation performed on the output of a differentially private algorithm does not cause any loss in privacy.

Theorem 3.4.1. (*Post-Processing*) *Let $\mathcal{A} : \mathcal{X} \mapsto R$ be a randomized algorithm that is (ϵ, δ)-DP. Let $f : R \mapsto R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{A} : \mathcal{X} \mapsto R'$ is (ϵ, δ)- DP.*



(a) Original (b) ϵ=1, δ=0.05, and d=2

(c) ϵ=1, δ=0.05, and d=5 (d) ϵ=1, δ=0.05, and d=10

Figure 3.3: The word cloud of the Facebook dataset visualizing the histogram as it changes after adding different levels of noise.

Another result is that the privacy of DP-mechanism can be amplified if it is preceded by a sampling step.

Theorem 3.4.2. *Let $\mathcal{A}$ be an (ϵ, δ)-DP algorithm and $\mathcal{D}$ is an input dataset. Let $\mathcal{A}'$ be another algorithm that runs $\mathcal{A}$ on a random subset of $\mathcal{D}$ obtained by sampling it with probability β. Algorithm $\mathcal{A}'$ will satisfy (ϵ', δ')-DP where $\epsilon' = \ln(1 + \beta(e^{\epsilon} - 1))$ and $\delta' < \beta\delta$.*

Additionally, we define a DP mechanism namely the truncated Laplace mechanism [129] which is used in Pr$\epsilon\epsilon$ch.

Definition 3.4.2 (Truncated Laplace mechanism for histogram)**.** Given a histogram H, the truncated Laplace mechanism, $\mathrm{Lp}(\epsilon, \delta, d)$, adds a non-negative integer noise vector $[\max(\eta, 0)]^{|\mathcal{V}|}$ to H, where η follows a distribution, denoted by $L(\epsilon, \delta, d)$ with a p.d.f $\Pr[\eta = x] = p \cdot e^{-(\epsilon/d)|x - \eta^0|}$, where $p = \frac{e^{\epsilon/d} - 1}{e^{\epsilon/d} + 1}$ and $\eta_0 = -\frac{d \cdot \ln((e^{\epsilon/d}+1)\delta)}{\epsilon} + d$.

Theorem 3.4.3. *The truncated Laplace mechanism satisfies (ϵ, δ)-DP for d-distant histogram releases [129].*

Fig. 3.3 visualizes the histogram of the Facebook dataset as a word cloud for different noise levels. As evident from the original word cloud, the histogram emphasizes few important words such as Facebook, people, information, and users. With

increased value of d, the resulting histogram has a roughly uniform distribution of the included words.

DP Discussion

Preech's use of DP is different from the most standard use-case of DP (like numeric datasets). It deals with concrete units like words instead of numeric statistics—introducing new challenges. We discuss these challenges and how Preech circumvents them in this section.

Vocabulary definition: The foremost task for defining the word histogram is defining the vocabulary, $\mathcal{V}$. The most conservative approach to define $\mathcal{V}$ is to consider the total set of all English stemmed and non-stop words. Such a vocabulary would be prohibitively large for efficient and practical usage. However, note that such a definition of $\mathcal{V}$ is an overestimate as no real-world document would contain all possible English words. Recall that our objective of adding noise is to obfuscate any statistical analysis built on top of the document's BoW (histogram), such as a topic modeling and stylometry analysis. Typically, BoW based statistical analyses are concerned only with the set of most frequent words. For example, any standard topic model captures only the top m percentile most frequent words in a transcript [120, 121]. The same applies to stylometry analysis, which is based on measures of the unique distribution of frequently used words of different individuals.

Thus, as long as the counts of the most common words of the transcript are protected (via DP), the subsequent statistical model (like topic model) built over the word histogram will be privacy-preserving too (by Thm. 3.4.1). However, high-frequency words might not be the only ones that contain important information about T_S. To tackle this, we also include words with large Term Frequency-Inverse Document Frequency (TF-IDF) weight to our vocabulary. This weight is a statistical measure used to evaluate how significant a word is to a document relative to a baseline corpus. The weight increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the baseline corpus. This offset adjusts for the fact that some words appear more frequently in general. To this end, Preech makes an estimate of the vocabulary from T_S^{OSP}. Although existing offline transcribers have high WER, we found (empirically) that they can identify the set of domain words of S with high accuracy (details in Sec. 3.7). For computing the TF-IDF values, IDF is computed using an external NLP

corpus like Wikipedia articles. Thus formally, $\mathcal{V} = \{w | w \in \{\text{top } \mathfrak{m} \text{ percentile of the most frequent words in } \mathsf{T}_S^{OSP}\} \cup \{\text{words with TF-IDF value} \geqslant \Delta \text{ in } \mathsf{T}_S^{OSP}\}\}$. Note that $\mathcal{V}$ should be devoid of all sensitive words which are scrubbed off from S in step 2 of Fig. 3.1. Additionally, the vocabulary can be extended to contain *out-of-domain* words, i.e., random English words that are not necessarily part of the original document. This helps in protecting against text classification attacks (Sec. 3.7).

Specificities of the word histogram: As discussed above, the goal of the DP mechanism is to generate noisy counts for each $w_i \in \mathcal{V}$. An artifact of our setting is that this noise has to be non-negative and integral. This is because dummy words (for the noisy counts) can only be added to S; removing any word from S is not feasible as this would entail in recognizing the word directly from S, which would require accurate transcription. Hence, Pr$\epsilon\epsilon$ch uses the truncated Laplace mechanism to ensure non-negative and integral noise.

Setting privacy parameters: The parameters ϵ and δ quantify the privacy provided by a DP-mechanism; the lower the values the higher the privacy guarantee achieved. The distance parameter d, intuitively, connects the privacy definition in the word histogram, which is purely a formal representation, to a semantic privacy notion. For example, it can quantify how much the noisy topic models computed by the CSP (from T_S^{CSP}) should differ from that of T_S^{g}. Thus, the user can tune d depending on the target statistical analysis. In the following, we detail a mechanism, as a guide for the user, for choosing d when the target statistical analysis is topic modeling.

Let us assume that the user has a set of speech files $\{S_j\}$ to be transcribed. Let D_j denote the ground truth transcript corresponding to speech file S_j. The objective is to learn t topics from the corpus $\bigcup_j D_j$ with at least k words per topic (a topic is a distribution over a subset of words from the corpus). Let $\mathcal{T} = \{\mathbb{T}_1, \cdots, \mathbb{T}_t\}$ represent the original topic model built on $\bigcup_j D_j = \bigcup_j \mathsf{T}_{S_j}^{g}$ and $\mathcal{T}' = \langle \mathbb{T}'_1, \cdots, \mathbb{T}'_t \rangle$ represent the noisy topic model computed by the CSP.

The following theorem (Thm. 3.4.4) provides a lower bound on the pairwise ℓ_1 distance between the true and noisy topics as a function of the privacy parameters of the DP word histogram release mechanism (specifically, the term C_{min} is a function of (d, ϵ, δ)).

Theorem 3.4.4. *For any pair of topics* $(\mathbb{T}, \mathbb{T}') \in \mathcal{T} \times \mathcal{T}'$,

$$||T - T'||_1 \geqslant 2\frac{1}{\left(1-(t-1)\frac{k}{\max_j |D_j|}\right)}\left(\frac{\mathcal{C}_{min}}{t} - \frac{1}{2}\left(1 - t\frac{k}{\max_j |D_j||}\right)\right),$$

where $\mathcal{C}_{min} = \min_{j,l}\left\{\frac{\nu\cdot(|D_j|-|w_{l,j}|\omega_j)}{|D_j|\cdot(|D_j|+\nu\cdot\omega_j}\right\}$, $|D_j|$ *is the total number of words in* D_j, ω_j *is the total number of unique words,* ν *is the variance of the distribution* $Lp(\epsilon', \delta', d)$, $\delta' = \beta\delta$ *and* $|w_{l,j}|$ *is the number of times the word* $w_l \in \mathcal{V}$ *appears in* D_j.

The proof of this theorem and the parameters description are in Appendix 7.3.

Dummy word injection: As discussed earlier, achieving differential privacy requires adding dummy words to S. Preech generates the dummy text corpus using an NLP language model (Sec. 3.6). The model takes in a short text sample from the required topic and generates an entire document of any required length based on that input. In some scenarios, the user can also provide a corpus of non-publicly available documents with the same vocabulary. This scenario is valid in many practical settings. For instance, in an educational institution, the sensitive speech files requiring transcription might be the interviews/oral exams of the students conducted on a specific subject, and the noise corpus can be the lecture notes of the same subject.

Next, Preech generates a set of dummy segments, $\mathbb{S}_d$, from the dummy corpus above. Let us assume that each of the true segments contains at most k non-stop words (depends on the segment length). Preech ensures that each dummy segment also contains no more than k non-stop words. Additionally, each such segment must contain only one word from the vocabulary $\mathcal{V}$. This means that although the physical noise addition is carried at the segment level, it is still equivalent to adding noise at the level of words (belonging to $\mathcal{V}$) as we only care about $w_i \in \mathcal{V}$. Each dummy segment is injected only once per CSP. Since the dummy segments have to be added in the speech domain, Preech applies TTS transforms to the segments in $\mathbb{S}_d$ such that they have the same acoustic features as $\mathbb{S}$. This condition ensures that $\mathbb{S}_d$ are indistinguishable from $\mathbb{S}$ in terms of their acoustic features. Preech provides the user with two broad options to satisfy this condition—voice cloning or voice conversion.

Voice cloning is a TTS system that generates speech in a target speaker voice. Given a speech sample from the target speaker, the system generates an embedding

of the speaker's voice biometric features. It uses this embedding to synthesize new utterances of any linguistic content in the target speaker's voice. Preech utilizes such a technology to clone the original speaker's voice and uses it to generate acoustically similar dummy segments $\mathbb{S}_d$. Preech applies a state-of-the-art voice cloning system [7], which generates a close-to-natural synthetic voice after being fine-tuned on a short sample (~ 5 sec.) from the target voice.

We evaluate this cloning system in Sec. 3.3, and the cloned samples are successfully identified as the true speakers. However, voice cloning does not protect the speakers' voice biometrics, and can be potentially thwarted by a stronger adversary. Hence, Preech provides voice conversion (VC) as a stronger privacy-preserving option for the user. VC transforms the voice of a source speaker to sound like a target speaker. Preech utilizes VC to obfuscate the true speakers' voice biometrics as well as to mitigate the DP noise indistinguishability concern by converting the true and dummy segments into a single target speaker voice (Sec. 3.4). We discuss the utility-privacy trade-offs of both options in Sec. 3.7.

It is important to note that the dummy segments do not affect the WER of T_S^{CSP}. It is so because Preech can exactly identify all such dummy segments (from their timestamps) and remove them from T_S^{CSP}. Additionally, since the transcription is done one segment at a time, the dummy segments do not affect the accuracy of the true segments ($\mathbb{S}$) either. Segmentation and voice conversion are the culprits behind the WER degradation, as will be evident in Sec. 3.7. Thus in Preech, the noise (in the form of dummy segments) can ensure differential privacy without affecting the utility. This is in contrast to standard usage of differential privacy for releasing numeric statistics where the noisy statistics result in a clear loss of accuracy. However, the addition of the dummy segments in Preech does increase the monetary cost of using the online service that has to transcribe more speech data than needed. We analyze this additional cost in Sec. 3.7.

In practice, we have multiple well-known cloud-based transcription services with low WER like Google Cloud Speech-to-Text, Amazon Transcribe, etc. Preech uses them to its advantage in the following way. Preech splits the set of segments $\mathbb{S}$ into N different sets (step 3 in Sec. 3.4) $\mathbb{S}_i, i \in [N]$ where N is the number of CSPs with low WER. Then, Preech sends each subset to a different CSP (after adding suitable noise segments to each set and shuffling them). Since each engine is owned by a different, often competing corporation, it is reasonable to assume that the CSPs are *non-colluding*. Thus, assuming that each segment contains at most

one word in $\mathcal{V}$, each subset of segments $\mathbb{S}_i$ can be viewed as randomly sampled sets from $\mathbb{S}$ with sampling probability $\beta = 1/N$. From Thm. 3.4.2, this partitioning results in a privacy amplification.

Mechanism

We summarize the DP mechanism by which Pr$\epsilon\epsilon$ch generates the dummy segments for S. The inputs for the mechanism are (1) $\mathbb{S}$ – the short segments of the speech file S, (2) the privacy parameters ϵ and δ and (3) N – the number of non-colluding CSPs to use. This mechanism works as follows:

- Identify the vocabulary $\mathcal{V} = \{w | w \in \{$ top m percentile of the most frequent words in $T_S^{OSP}\} \cup \{$ words with TF-IDF value $\geqslant \Delta$ in $T_S^{OSP}\}\}$ through running an offline transcriber over S.
- Tune the value of d based on the lower bound from Thm. 3.4.4, ϵ and δ.
- Generate N separate noise vectors, $\eta_i \sim [Lp((\ln(1+\frac{1}{\beta}(e^{\epsilon}-1)), \beta\delta, d)]^{|\mathcal{V}|}, i \in [N]$. Thus for every partition i, Pr$\epsilon\epsilon$ch associates each word in $\mathcal{V}$ with a noise value, a non-negative integer.
- From the NLP generated text, extract all the text segments that contain words from $\mathcal{V}$. For each partition i, sample the text segments from this corpus to match the noise vector η_i. This is the set of noise (dummy) segments for partition i, $\mathbb{S}_{d,i}$. Iterate on generating text from the NLP language model until the required noise count is satisfied.
- Randomly partition $\mathbb{S}$ into N sets $\mathbb{S}_i, i \in [N]$ where Pr[segment s goes to partition i] $= \beta = 1/N, s \in \mathbb{S}$.
- For each partition $i \in [N]$, shuffle the dummy segments in $\mathbb{S}_{d,i}$ (after applying TTS and VC) with the segments in $\mathbb{S}_i$ (after applying VC), and send it to the CSP_i.

The first 4 steps in the above mechanism are performed in stage 3 in Pr$\epsilon\epsilon$ch (Fig. 3.1) while steps 5-6 are performed in stage 6.

Theorem 3.4.5. *Any topic model computed by* $CSP_i, i \in [N]$ *from* $T_S^{CSP_i}$ *is* (ϵ, δ)*-DP.*

Proof. From Thm. 3.4.2 and Thm. 3.4.3, we conclude that the word histogram $\tilde{H}_i$ computed from $T_S^{CSP_i}$ is (ϵ, δ) - DP for distance d. Thm. 3.4.1 proves that the topic model from $\tilde{H}_i$ is still (ϵ, δ)-DP as it is a post-processing computation. □

Novelty of Preech's Use of Differential Privacy

Here, we summarize the key novelty in Preech's use of DP:
(1) Typically, DP is applied to statistical analysis of numerical data where "noise" corresponds to numeric values. In contrast, in Preech, "noise" corresponds to concrete units – words. To tackle this challenge, we applied a series of operations (segmentation, shuffling, and partitioning) to transform the speech transcription into a BoW model, where the DP guarantee can be achieved. Moreover, the noise addition has to be done in the speech domain. This constraint results in new challenges: the lack of a priori access to the word histogram domain $\mathcal{V}$, and generating indistinguishable dummy speech segments.
(2) In our setting, the use of a DP mechanism does not introduce a privacy-utility trade-off from the speech transcription standpoint. Preech performs transcription one segment at a time. It keeps track of the timestamps of the dummy segments and completely removes their corresponding text from the final transcription (Sec. 3.4). This filtration step is achievable in Preech, unlike numeric applications of DP, because of the atomic nature of transcription. However, the dummy segments increase the monetary cost of transcription, resulting in a privacy-monetary cost trade-off as shown in Table 3.3. To tackle this issue, Preech takes advantage of the presence of multiple CSPs (Sec. 3.4). Thus, the idea of utilizing multiple CSPs for cost reduction (Thm. 3.4.2) is a novel contribution.
(3) We introduce an additional parameter d, the distance between the pair of histograms, in our privacy definition (Def. 3.4.1). Intuitively, d connects the privacy definition in the word histogram model, which is purely a formal representation, to a semantic privacy notion (e.g., ℓ_1 distance between true and noisy topic models, Thm. 3.4.4) as shown in Fig. 3.7 and 3.6. This contribution builds on ideas like group privacy [110] and generalized distance metrics [130].

Control Knobs

The construction of the DP word histogram provides the user with multiple control knobs for customization:
Parameter d: According to Def. 3.4.1, from $\tilde{H}$ the CSP will not be able to distinguish between T_S^{CSP} and any other transcript that differs from T_S^{CSP} in d words from $\mathcal{V}$. Thus, higher the value of d, larger is the ϵ-indistinguishability neighborhood for $\tilde{H}$ and hence, better is the privacy guarantee. But it results in an increased amount of

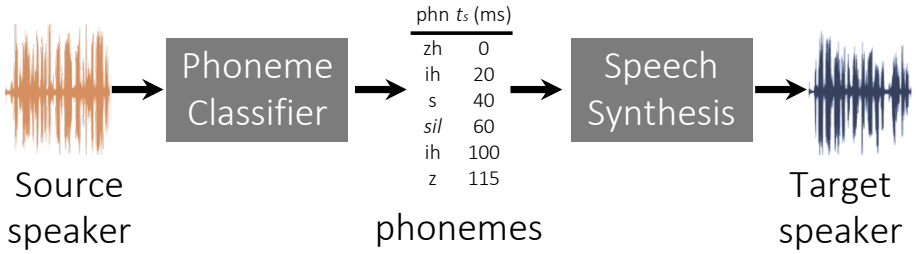


Figure 3.4: An illustration of the many-to-one VC pipeline.

noise injection (hence, increased monetary cost – details in Sec. 3.7).

Vocabulary: The size of $\mathcal{V}$ is a control knob, specifically, the parameters $\mathfrak{m}$ and Δ and the number of *out-of-domain* words. The trade-off here is: the larger the size of $\mathcal{V}$, the greater is the scope of the privacy guarantee. However, the noise size scales with $|\mathcal{V}|$ and hence incurs higher cost (details in Sec. 3.7).

Voice transformation for noisy segments: Preech provides two options for noise synthesis – voice cloning and voice conversion. Voice cloning does not affect the transcription utility, measured in WER, because it does not apply any transformations on the original speaker's voice. However, it fails to protect the sensitive biometric information in S. Moreover, there is no guarantee that a strong adversary cannot develop a system that can distinguish the cloned speech segments from the original ones. This puts Preech's effectiveness at the risk of the arms race between the voice cloning system's performance and the adversary's strength. This limitation is addressed by voice conversion at the cost of transcription utility. We quantify these utility-privacy trade-offs in Sec. 3.7.

Number of CSPs used for transcription: As discussed above, employing multiple CSPs lowers the monetary cost incurred. However, as shown in Table 3.1, AWS has a higher WER than Google. Hence, using both the CSPs results in lower overall utility than just using Google's cloud service.

Voice Conversion

Below, we discuss the two main categories of VC systems, highlighting their privacy-utility trade-offs.

One-to-One Voice Conversion

One-to-one VC maps a predefined source speaker voice to a target speaker voice. In Preech, we use sprocket [123], which is based on spectral conversion using a Gaussian mixture model (GMM). Sprocket's training phase takes three steps: (1) acoustic features extraction of the source and target speakers samples, (2) time-alignment of the source and target features, and (3) GMM model training. During conversion, sprocket extracts the acoustic features of the new utterances, converts them using the learned GMM model, and generates the target waveform. Preech applies sprocket to convert the voice of all source speakers, including the synthesized dummy segments, into the *same target speaker voice*.

Many-to-One Voice Conversion

For perfect voice privacy, the VC system should (1) map any voice (even if previously unseen) to the same target voice, (2) not leak any distinguishing acoustic features, and (3) operate on speech containing multiple speakers. To this end, Preech deploys the two-stage many-to-one VC [124] mechanism. As shown in Fig. 3.4, the first stage is a phoneme classifier that transfers the speech utterance into a phonetic posterior grams (PPG) matrix. A PPG is a time-aligned phonetic class [124], where a phoneme is the visual representation of a speech sound. Thus, the phoneme classifier removes the speaker-identifying acoustic features by mapping the spoken content into speaker-independent labels. In the second stage, a speech synthesizer converts the PPGs into the target voice.

The PPGs intermediate stage is irreversible and speaker-independent. It guarantees that the converted dummy segments $\mathbb{S}_d$ and converted original segments $\mathbb{S}$ cannot be distinguished from each other. However, the actual implementation of the system carries many challenges. The first stage is a performance bottle-neck as it needs large phonetically aligned training data to generalize to new unseen voices. We overcome this challenge by generating a custom training speech dataset with aligned phonemes as described in Sec. 3.6.

Control Knobs

The aforementioned VC techniques present an interesting utility-usability-privacy trade-off. The one-to-one VC technique gives better accuracy than many-to-one VC

since it is trained for a specific predefined set of source speakers (details in Sec. 3.7). However, this utility gain comes at the price of usability and privacy. First, unlike many-to-one VC, sprocket needs parallel training data—a set of utterances spoken by both the source and target speakers. Hence, it requires an enrollment phase to get the source speaker's voice samples, thereby limiting the scalability of Preech for previously unseen speakers. Second, one-to-one VC does not provide perfect indistinguishability. These two limitations can be mitigated by applying many-to-one VC (Sec. 3.7) at the expense of degraded transcription accuracy.

3.5 End-to-End Threat Analysis

In this section, we go over the end-to-end system design of Preech and identify potential privacy vulnerabilities.

Voice Privacy: Many-to-one VC removes all the identifying features from S, like the speakers' voices, background noise, and recording hardware, thereby protecting voice privacy.

Textual Privacy: For sensitive word scrubbing, the best-case scenario from a privacy point of view is to have the user spell out the entire keyword list. However, due to its high usability overhead, Preech uses NER instead to identify named entities automatically from T_S^{OSP}. In Sec. 3.7, we empirically show that Preech can achieve a near-perfect true positive rate in identifying the segments containing sensitive words. However, this is only an empirical result and is dataset dependent.

Our main defense against statistical analysis on the text is the DP guarantee on the word histogram. This DP guarantee would break down if the adversary can distinguish the dummy segments from the true segments. Many-to-one VC technique, by design, ensures that both sets of segments have the same acoustic features. However, the possibility of distinguishing them based on their textual features still remains. To address this threat, we rely on state-of-the-art NLP models with low perplexity (log-likelihood) scores to generate the dummy text corpus. The low perplexity scores ensure that the auto-generated text is as close as possible to the natural language generated by humans [131, 132]. Although there is no formal guarantee about the adversary's ability to distinguish dummy and true

segments based on their textual features, we have empirically analyzed this threat in Sec. 3.7 and Sec. 3.7. We leverage state-of-the-art NLP techniques to mount attacks on the dummy segments. Our results show that the adversary fails to distinguish between the dummy and true segments. However, the extent of such robustness is based on the efficacy of state-of-the-art NLP techniques.

Word correlations can also weaken the DP guarantee ($d - w$, if w is the maximum size of word groups with high correlation). This can be addressed by either increasing d or considering n-gram ($n = w$) word histograms. However, this would increase the requisite amount of dummy segments.

Long segments can also be a source of privacy vulnerability as each segment contains more contextual information. Hence, in the prototype Preech presented in the chapter, we use short segments that contain at most two non-stop words.

Another weakness is related to vocabulary estimation, especially if some of the distribution-tail words are deemed to be sensitive. Preech provides no formal guarantees on the words that do not belong to $\mathcal{V}$. Although our empirical evaluation shows that the OSP has a very high accuracy for the weighted estimation of $\mathcal{V}$ (Sec. 3.7), some sensitive distribution-tail words might still be missed due to the OSP's transcription errors. Additionally, our formal DP guarantee holds only for the word histogram (BOW) on $\mathcal{V}$. Textual analysis models other than BOW are empirically evaluated in Sec. 3.7 and Sec. 3.7.

Finally, if the CSP can reorder the segments (even partially since the speech file it receives contains dummy segments as well), it will be able to distinguish the dummy segments from the true ones and hence, learn the textual content of the file. For this again, we show empirically that current NLP techniques fail to reorder the segments (Sec. 3.7) even in the worst-case setting where all the segments go to one CSP. However, as before, this is an empirical result only.

Formal Privacy Guarantee: *For a speech file S,* Preech *provides perfect voice privacy (when using many-to-one VC) and an (ϵ, δ)-DP guarantee on the word histogram for the vocabulary considered* (BOW)*, under the assumption that the dummy segments are indistinguishable from the true segments.*

3.6 Implementation

In this section, we describe the implementation details of Preech's building blocks (shown in Fig. 3.1).

Segmentation: We implement the two-level hierarchical segmentation algorithm described in Sec. 3.4. The silence detection based segmentation is implemented using the Python pydub package[8]. We used Praat[9] to extract the pitch information required for the second level of the segmentation algorithm.

Sensitive Keyword Scrubbing: We use the NLP Python framework spaCy [10] for named entity recognition (NER) from the text. The keyword lists per each dataset can be found in Appendix 7.4. We employ PocketSphinx[11] for keyword spotting, a lightweight ASR that can detect keywords from continuous speech. It takes a list of words (in the text) and their respective sensitivity thresholds and returns segments that contain speech matching the words. PocketSphinx is a generic system that can detect any keyword specified in runtime; it is not trained on a pre-defined list of keywords and requires no per-user training or enrollment.

Generating Dummy Segments: We use the open source implementation [12] of OpenAI's state-of-the-art NLP language model, GPT2[132], to generate the noise corpus.

Using this predictive model, we generate a large corpus representing the vocabulary of the evaluation datasets. An example of the generated text is available in Appendix 7.5. To generate the dummy segments, we segment each document at the same level as the speech segmentation algorithm. We build a hash table associating each vocabulary word with the segments that contain it. Preech uses a dummy segment only once per CSP to prevent it from identifying repetitions.

Text-to-Speech: We use the multi-speaker (voice cloning) TTS synthesizer [7] to generate the speech files corresponding to the dummy segments. We use a

[8] https://pypi.org/project/pydub/
[9] http://www.fon.hum.uva.nl/praat/
[10] https://github.com/explosion/spaCy
[11] https://github.com/cmusphinx/pocketsphinx
[12] https://github.com/huggingface/transformers

pre-existing system implementation and pretrained models [13].

One-to-One Voice Conversion: We use the open-source sprocket software [14]. As described in Sec. 3.4, sprocket requires a parallel training data and the target voice should be unified for all source speakers. For the VCTK datasets, we use speaker p306 as the target voice. Since we also evaluate Preech on non-standard datasets (Facebook and Carpenter cases), we had to construct the parallel training data for their source speakers. For this, we use TTS to generate the required target voice training utterances in a single *synthetic* voice.

Many-to-One Voice Conversion: We utilize pre-existing architectures and hyperparameters [15] for the two-stage many-to-one VC [124] mechanism, shown in Fig. 3.4. The first network, net_1, is trained on a set of {raw speech, aligned phoneme labels} samples from a multi-speaker corpus, where the labels are the set of 61 phonemes from the TIMIT dataset. The only corpus that has a manual transcription of speech to the phonemes' level is the TIMIT dataset—a limited dataset. We found that training net_1 on TIMIT alone results in an inferior WER performance. For better generalization, we augment the training set by automatically generating phoneme-aligned transcriptions of standard ASR corpora. We use the Montreal Forced Aligner [16] to generate the aligned phonemes on LibriSpeech and TED-LIUM [133] datasets. The second network, net_2, synthesizes the phonemes into the target speaker's voice. It is trained on a set of {PPGs, raw speech} pairs from the target speaker's voice. We use the *trained* net_1 to generate the PPGs data for training net_2. As such, we only need speech samples of the target speaker to train net_2. This procedure also allows net_2 to account for net_1's errors. We use Ljspeech[17] as the target voice for its relatively large size—24 hours of speech from a single female.

[13] https://github.com/CorentinJ/Real-Time-Voice-Cloning
[14] https://github.com/k2kobayashi/sprocket
[15] https://github.com/andabi/deep-voice-conversion
[16] https://montreal-forced-aligner.readthedocs.io/en/latest/
[17] https://keithito.com/LJ-Speech-Dataset/

3.7 Preech Evaluation

We evaluate how well Preech meets the design objectives of Sec. 3.4. Specifically, we aim to answer the following questions:
(*Q1.*) Does Preech preserve the transcription utility?
(*Q2.*) Does Preech protect the speakers' voice biometrics?
(*Q3.*) Does Preech protect the textual content of the speech?
(*Q4.*) Does the different control knobs provide substantial flexibility in the utility-usability-privacy spectrum?

We answer the first three questions for a prototype implementation of Preech that provides the maximum degree of formal privacy and hence, the least utility. For evaluating *Q4*, we relax the privacy guarantee to obtain utility and usability improvements.

Prototype Preech: For the prototype Preech presented in the chapter: (1) segmentation length is adjusted to ensure that each segment contains at most two non-stop words (2) noisy segments are generated via the GPT2 language model (3) a single CSP (Google) is utilized (4) many-to-one VC is applied to both the dummy and true segments.

Q1. Transcription Utility

We assess the transcription WER after deploying end-to-end Preech on the non-standard datasets. Recall that Table 3.1 in Sec. 3.2 shows the baseline WER performance of the CSP and OSP before applying Preech.

WER Analysis: Column 4 in Table 3.2 shows the end-to-end WER for the prototype Preech which represents the accumulative effect of segmentation, SWS, and many-to-one VC. Although VC is the main contributor to Preech's WER, as is evident from Sec. 3.7, there are two main observations. First, many-to-one VC is superior to Deep Speech. Specifically, Preech's relative improvement over Deep Speech ranges from 11.91% to 32.25% over the evaluation datasets (except for Carpenter2). Recall that we trained the VC system using standard ASR corpora, while we evaluate the WER on non-standard cases. Still, Preech's WER is superior to that of Deep Speech, which has been trained through hundreds of hours of speech data. Second, Preech

Datasets	Cloning	One-to-One	Many-to-One	OSP
VCTK p266	5.15 (80.73%)	16.55 (38.06%)	21.92 (17.96%)	26.72
VCTK p262	4.53 (71.63%)	7.39 (53.73%)	10.82 (32.25%)	15.97
Facebook1	8.26 (66.59%)	14.60 (40.94%)	20.30 (17.88%)	24.72
Facebook2	9.75 (63.36%)	18.27 (31.34%)	19.44 (26.94%)	26.61
Facebook3	14.93 (51.40%)	23.25 (24.32%)	27.06 (11.91%)	30.72
Carpenter1	14.43 (44.18%)	23.88 (7.62%)	22.63 (12.46%)	25.85
Carpenter2	13.53 (65.93%)	33.71 (15.11%)	38.90 (2.04%)	39.71

Table 3.2: WER (%) of end-to-end Preech which represents the accumulative effect of segmentation, SWS, and different settings of voice privacy and its relative improvement in (%) over OSP (Deep Speech).

does not have the same performance for all the datasets. This observation arises again from the lack of diversity in our VC training set. For example, the speaker in Carpenter 1 speaks loudly, allowing VC to perform well. On the other hand, the second speaker (Carpenter 2) is not as clear or loud, which results in an inferior VC performance. This observation is consistent with Deep Speech as well.
Our experiments show that these results can be improved by adding samples of the source speaker voice to the training pipeline of net_1 and net_2. We chose not to go with this approach as this limits the usability of the system, and in such a case sprocket (Sec. 3.7) would be a better choice.

Q2. Voice Biometric Privacy

To test the voice biometric privacy, we conduct two experiments using the voice analysis APIs (details in Sec. 3.3). In the first experiment, we assess the CSP's ability to separate speech belonging to different speakers after Preech applies the VC system. On our multi-speaker datasets, IBM diarization API concludes that there is only one speaker present.

Furthermore, we run the diarization API after adding the dummy segments (after TTS and VC). Again, the API detects the presence of only one speaker. Thus, not only does Preech hide the speaker's biometrics and map them to a single target speaker but also ensures noise indistinguishability, which is key to its privacy properties.

The second experiment tests Preech's privacy properties against a stronger

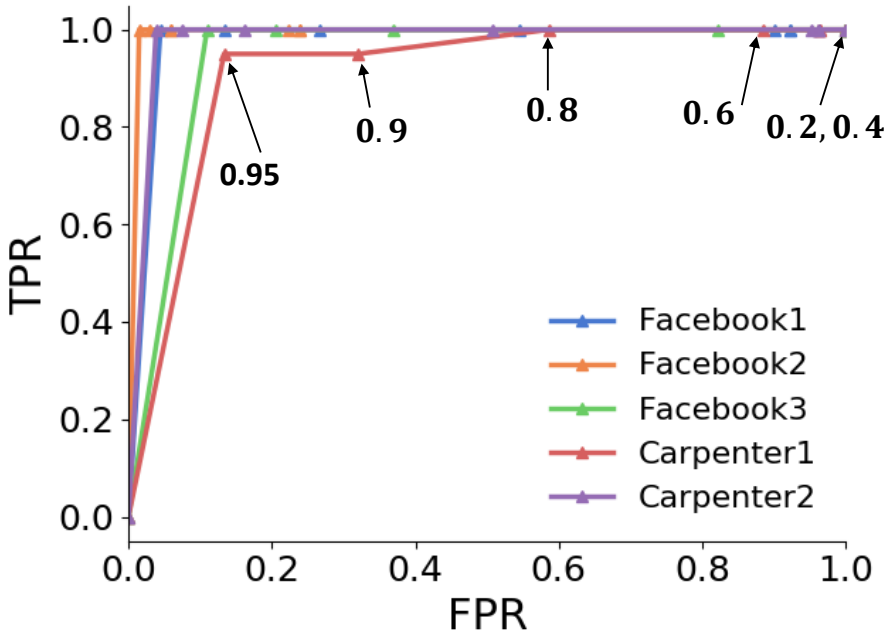


Figure 3.5: ROC curve for sensitive words detection at different values of the sensitivity score.

adversary, who has access to samples from the true speakers. We enroll segments from the true speakers as well as the fake target speaker to Azure's Speaker Identification API. We pass the segments from Preech (after adding dummy segments and applying VC) to the API. When many-to-one VC is applied, in all evaluation cases, the API identifies the segments as belonging to the fake target speaker. Not a single segment was matched to the original speaker. Both experiments show that prototype Preech is effective in sanitizing the speaker's voice and ensuring noise indistinguishability.

Q3. Textual Privacy

We perform an extensive evaluation of the textual privacy, including sensitive word scrubbing, analysis of the DP mechanism, and defense against statistical analysis.

Sensitive Words Scrubbing:

We run PocketSphinx keyword spotting on each dataset at different sensitivity scores ranging from 0.2 to 1[18]. Fig. 3.5 shows the detection true positive rate (TPR) versus the false positive rate (FPR) at different sensitivity scores. As the figure shows, the sensitivity score is a trade-off knob between privacy (high TPR) and utility (low FPR). We observe that Preech is able to achieve almost perfect TPR with low FPR values.

Next, we evaluate the impact of SWS on the transcription utility. We set a sensitivity score of 0.95 for all the datasets to have a near-perfect TPR while minimizing the FPR. Our experiments show that the total duration of the segments flagged with sensitive keywords at this score is: 0.13%, 0.06%, 0.18%, 0.20%, and 0.08% of the total duration of each dataset in Fig. 3.5. Then, we transcribe the sensitive-flagged segments using Deep Speech. The overall transcription accuracy after SWS (i.e., equivalent to choosing voice cloning in Preech as cloning results in no addition WER) is presented in the second column of Table 3.2. Since the segments are short, the portion of speech transcribed locally is limited. Hence, the impact of the OSP transcription errors is not significant.

DP Mechanism Analysis:

We follow the DP mechanism described in Sec. 3.4.

Vocabulary Estimation: We estimate the vocabulary $\mathcal{V}$ using the OSP transcript. Let $\mathcal{W}$ represent the set of unique words in T_S^g. We define the accuracy of the vocabulary estimation, D_{acc}, as the ratio between the count of the correctly identified unique words from T_S^{OSP}, $|\mathcal{W}|_{est}$, and the count of the unique words in T_S^g, $|\mathcal{W}|$. For our datasets, the domain estimation accuracy is at least 75.54%. We also calculate the weighted estimation accuracy defined as: $D_{weighted} = \frac{\sum P(w_{est}).\mathbb{1}_{w_{est} \in \mathcal{W}}}{|\mathcal{W}|}$ where $P(w_{est})$ is the weight of the estimated word w_{est} in T_S^g. $D_{weighted}$ is more informative since it gives higher weights to the most frequent words in T_S^g. The weighted estimation accuracy is 99.989% in our datasets. From $\mathcal{W}_{est}$ we select $\mathcal{V}$ over which we apply the DP mechanism. Additionally, we extend our vocabulary to contain a set of random words from the English dictionary.

[18]The sensitive keywords list for each dataset is in Appendix 7.4.

Datasets	$\|\mathcal{V}\|$	# words in T_S^g	#Extra words due to dummy segments d=2	d=5	d=15
VCTK p266	483	922 ($0.22)	2915 ($0.68)	7247 ($1.69)	23899 ($5.58)
VCTK p262	471	914 ($0.21)	2845 ($0.66)	7157 ($1.67)	23230 ($5.42)
Facebook	1098	5326 ($1.24)	6660 ($1.55)	16567 ($3.87)	54038 ($12.62)
Carpenter	1474	7703 ($1.80)	8915 ($2.08)	22296 ($5.20)	72907 ($17.02)

Table 3.3: Number of extra words due to dummy segments and the additional monetary cost in USD with varying d, at $\epsilon = 1$ and $\delta = 0.05$.

Histogram Distance: We analyze the distance between the original and noisy histograms (after applying Preech) and its impact on the cost of online transcription. Because of the nature of Preech's DP mechanism, the noise addition depends on four values only: $|\mathcal{V}|$, ϵ, δ, and d.

For all our experiments, we fix the values of $\epsilon = 1$ and $\delta = 0.05$. Table 3.3 shows the amount of noise (dummy words) and their transcription cost in USD [19] for each of the evaluation datasets at different values of d. Each dataset has a different vocabulary size $|\mathcal{V}|$ and word count. The increase in the vocabulary size requires adding more dummy segments to maintain the same privacy level. In Preech, adding more noise comes at an increased monetary cost, instead of a utility loss. The table highlights the *trade-off* between privacy and the cost of adding noise.

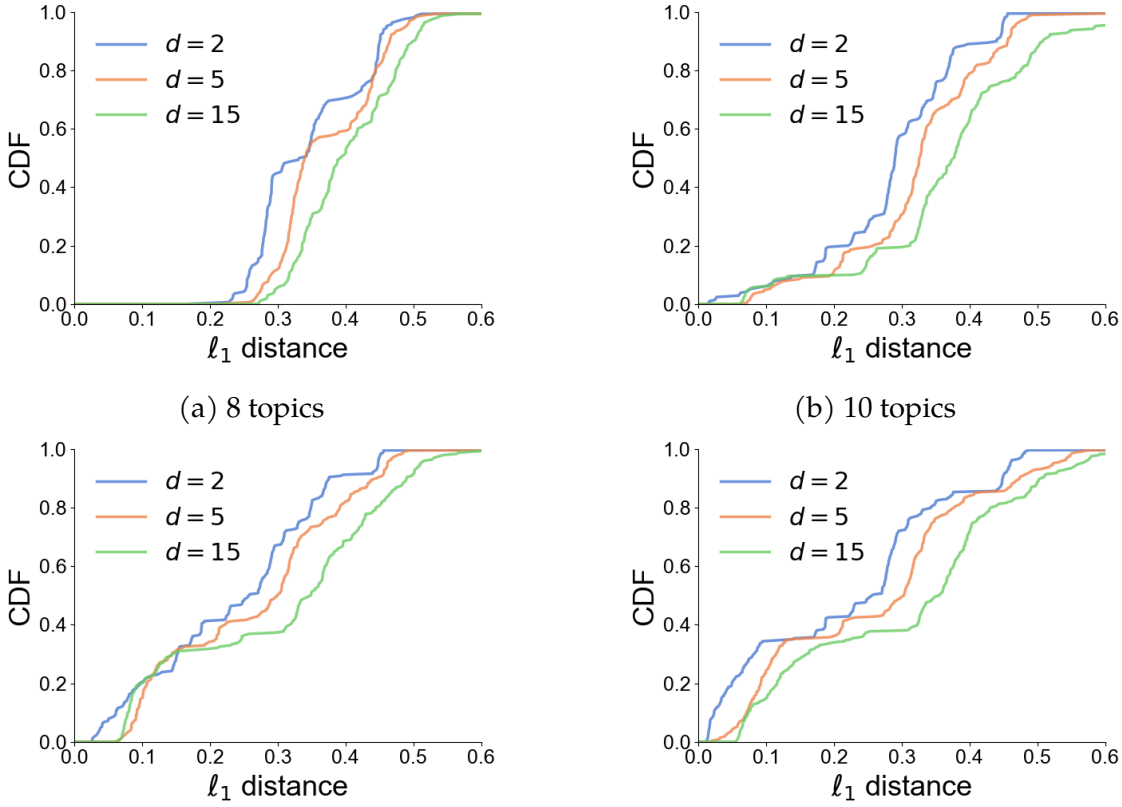
Statistical Analysis

In this section, we evaluate the statistical analyses (details in Sec. 3.3) performed by the adversary to extract textual information on the noisy transcripts obtained from Preech.

Topic Model: We generate the topic models from the documents corresponding to the original and noisy word histograms, and evaluate their ℓ_1 distance. The topic model operates on a corpus of documents; hence we include eight more Supreme Court cases to our original evaluation datasets (Facebook and Carpenter). In this evaluation, we treat all these ten documents as one corpus; we aim to generate the topic model before and after applying Preech to the whole corpus.

We use AWS Comprehend API to generate the topic model. The API needs the number of topics as a hyperparameter that ranges from 1 to 100. Based on our

[19] The pricing model of Google Speech-to-Text is: $0.009/15 seconds.



(a) 8 topics

(b) 10 topics

(c) 12 topics

(d) 14 topics

Figure 3.6: Topics ℓ_1 distance CDF at d = 2, 5, and 15 for t = 8, 10, 12, and 14

apriori knowledge of the true number of topics, we evaluate the topic model on the following number of topics $t = 8, 10, 12$, and 14.

We statistically evaluate the ℓ_1 distance between true and noisy topics. The topic model $\mathcal{T} = \{\mathbb{T}_1, \cdots, \mathbb{T}_t\}$ is a set of t topics where each $\mathbb{T}_i, i \in [t]$ is a word distribution. We use the Hungarian algorithm to match each noisy topic $\mathbb{T}'_i \in \mathcal{T}'$ to its closest match in $\mathcal{T}$, the true topic model. We evaluate the topics ℓ_1 distance for 21 runs. At each run, we generate a random noise vector per document, select the corresponding dummy segments, and evaluate the topic model on the set of original and noisy documents. Fig. 3.6 shows the empirical CDF of the topics ℓ_1 distance at different values of d. As the figure shows, the higher the distance parameter d, the larger is the ℓ_1 distance between true and noisy topics.

Stylometry: In this experiment, we assume that the CSP applies stylometry analysis on T_S^{CSP} in an attempt to attribute it to an auxiliary document whose

authors are known to the CSP. To evaluate the worst-case scenario, we assume the adversary possesses the original document T_S^g, and we compute the ℓ_2 distance of the stylometric feature vectors generated from T_S^{CSP} w.r.t T_S^g.

First, we compute the ℓ_2 distance of T_S^{CSP} before applying Pr$\epsilon\epsilon$ch. The respective values for the Facebook and Carpenter datasets are 28.19 and 60.45. T_S^{CSP} differs from T_S^g in lexical features due to transcription errors and because the CSP generates the punctuation instead of the actual author.
Second, we apply Pr$\epsilon\epsilon$ch on the two datasets at different values of the distance parameter: $d = 0, 2, 5, 15$. The corresponding ℓ_2 distances for the Facebook (Carpenter) dataset equal: 73.14 (83.64), 328.80 (577.72), 947.58 (1629.79), and 2071.18 (3582.10). Note that the ℓ_2 distance at $d = 0$ shows the effect of segmentation and SWS only on obfuscating the lexical features. Clearly, adding the dummy segments increases the ℓ_2 distance. This is expected as most of the lexical features are obfuscated by the DP mechanism.

Category Classification: Google's NLP API can classify a document to a predefined list of 700+ document categories[20]. First, we run the classification API on the original documents from the topic modeling corpus. All of them classify as Law & Government. Running the API on Pr$\epsilon\epsilon$ch processed documents, using an extended vocabulary (i.e., contains random words), dropped the classification accuracy to 0%. None of the documents got identified as legal, law, or government even at the smallest distance parameter value $d = 2$. Although a portion of the noise words belongs to the original Law & Government category, segmentation, shuffling, and the out-of-domain noise words successfully confuse the classifier.

Sentiment Analysis: Sentiment analysis generates a score in the $[-1, 1]$ range, which reflects the positive, negative, or neutral attitude in the text. First, we evaluate the sentiment scores of the original ten documents. For all of them, the score falls between -0.2 and -0.9, which is expected as they represent legal documents. Next, we evaluate the scores from Pr$\epsilon\epsilon$ch processed documents considering an extended-vocabulary. We find that all scores increase towards a more positive opinion. Fig. 3.7 shows a heatmap of the sentiment scores as we change the distance parameter d for the ten evaluation documents. Thus, Pr$\epsilon\epsilon$ch's two-pronged approach—1) addition of extended-vocabulary noise, and 2) removal of ordering information

[20] https://cloud.google.com/natural-language/docs/categories

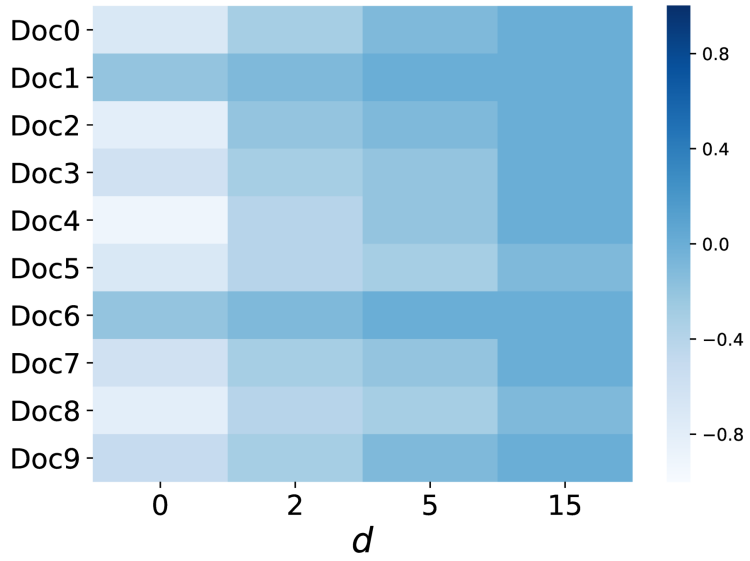


Figure 3.7: Sentiment scores heatmap of 10 documents with varying d, at $\epsilon = 1$ and $\delta = 0.05$.

via segmentation and shuffling—proves to be effective. In a setting where the adversary has no apriori knowledge about the general domain of the processed speech, the noise addition mechanism gains extend from DP guarantee over the histogram to other NLP analyses as well.

Indistinguishability Of Dummy Segments

The indistinguishability of the dummy segments is critical for upholding the DP guarantee in Prεεch. We perform two experiments to analyze whether current state-of-the-art NLP models can distinguish the dummy segments from their textual content.

Most Probable Next Segment: In this experiment, the adversary has the advantage of knowing a true segment $\mathbb{S}_t$ that is at least a few sentences long from the Facebook dataset. We use the state-of-the-art GPT [21] language model by OpenAI [134] to determine the most probable next segment following $\mathbb{S}_t$ using the model's perplexity score. In NLP, the perplexity score measures the likelihood that a piece of text follows the language model. We get the perplexity score of stitching $\mathbb{S}_t$ to each of the other segments at the CSP. The segment with the lowest

[21] `https://github.com/huggingface/transformers`

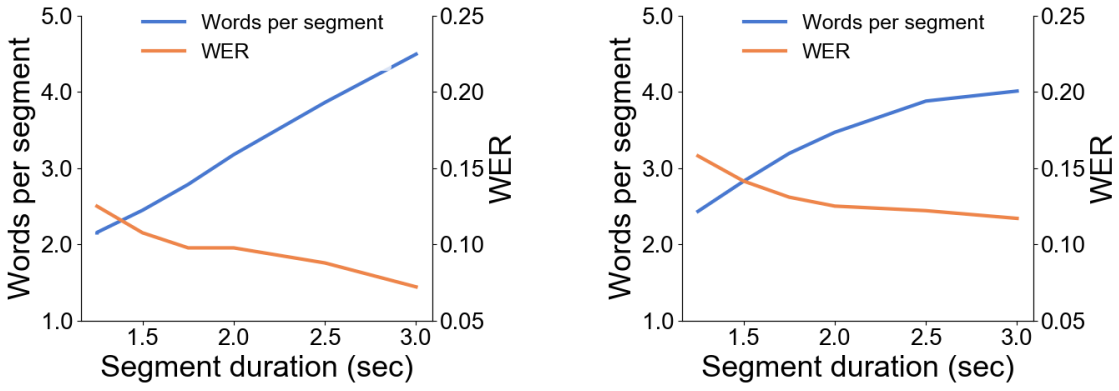
perplexity score is selected as the most probable next segment. We iterate over all the true segments of the Facebook dataset, selecting them as $\mathbb{S}_t$. We observed that a dummy segment is selected as the most probable next segment in 53.84% of the cases. This result shows that the language model could not differentiate between the true and dummy segments even when part of the true text is known to the adversary.

Segments Re-ordering: Next, we attempt to re-order the segments based on the perplexity score. We give the adversary the advantage of knowing the first true segment $\mathbb{S}_0$. We get the perplexity score of $\mathbb{S}_0$, followed by each of the other segments. The segment with the lowest score is selected as the second segment $\mathbb{S}_1$ and so on. We use the normalized Kendall tau rank distance K_τ to measure the sorted-ness of the re-ordered segments. The normalized K_τ distance measures the number of pairwise disagreements between two ranking lists, where 0 means perfect sorting, and 1 means the lists are reversed. The K_τ score for running this experiment on the Facebook dataset is 0.512, which means that the re-ordered list is randomly shuffled w.r.t the true order. Hence, our attempt to re-order the segments has failed.
These empirical results show that it is hard to re-order the segments or distinguish the dummy segments. This is expected due to three reasons: (1) the segments are very short; (2) the dummy segments are generated using a state-of-the-art language model; and (3) we observed that most of the transcription errors happen in the first and last words of a segment due to breaking the context. These errors add to the difficulty of re-ordering. Moreover, if the user partitions S among multiple CSP's (Sec.3.4), then consecutive segments would not go to the same CSP with high probability. This setting would increase Preech's protection against re-ordering attacks.

Q4: Flexibility of the Control Knobs

Utility-Privacy Trade-off

In this section, we empirically evaluate the control knobs that provide a utility-privacy trade-off.



(a) Facebook (b) Carpenter

Figure 3.8: Segmentation trade-off between utility and privacy. WER(%) is measured using Google Cloud Speech-to-Text.

Minimum segment length: Fig. 3.8 shows the trade-off between the number of words per segment and WER as function of the minimum segment length. As expected, increasing the minimum duration of a segment results in an increase in the number of words per segment. The WER in turn drops when the number of words per segment increase as the transcription service has more textual context. However, it can lead to potential privacy leakage. The results in Fig. 3.8 indicate that for two real-world datasets, the number of words per segment can be kept between 2 and 3 with an acceptable degradation of the WER.

Voice Cloning: Voice cloning does not affect the true segments (it is only applied to dummy segments), resulting in no additional WER degradation. The WER for deploying voice cloning is incurred only due to segmentation and SWS. Thus, as shown in column 2 of Table 3.2, the relative improvement in WER ranges from 44% to 80% over Deep Speech. This approach, however, has two limitations. First, the speaker's voice biometrics from S are not protected. Second, there is no guarantee that an adversary would not be able to distinguish the cloned speech segments from the original ones.

Sensitivity score of KWS: As shown in Fig. 3.5, the lower the sensitivity score, the higher the TPR and hence the greater the privacy (most prominent in the Carpenter2 dataset). However, this also increases the FPR, which means a larger number of non-sensitive segments are transcribed via the OSP resulting in reduced

accuracy.

One-To-One VC: Table 3.2, column 3, shows that one-to-one VC outperforms many-to-one VC on most of the datasets. This result is expected since sprocket is trained and tested on the same set of source speakers while the many-to-one VC system generalizes to previously unseen speakers.
We observe that the improvement for the VCTK dataset is more significant than others. Recall that in our one-to-one VC implementation in Sec. 3.6, the target voice for VCTK is a natural voice—speaker p306. The target voice for the other datasets is a synthetic one, which hinders the quality of the converted voice and the transcription accuracy. We investigate this observation by training sprocket for VCTK on a synthetic target voice as well. The WER then increased to 19.33% and 9.21% for p266 and p262. Hence, we attribute the difference in the relative improvement to the target voice naturalness. In practice, the target voice could easily be a natural pre-recorded voice, and the users are asked to repeat the same utterances at the enrollment phase.
However, the one-to-one VC technique suffers from some privacy loss. The one-to-one VC system translates the acoustic features from a source to a target speaker's voice. Hence, it may leak some features from the source speaker. We observed that one-to-one VC is vulnerable to speaker identification analysis. Specifically, using Azure's Speaker Identification API, 10% of the voice-converted segments using sprocket were identified to their true speakers.

Usability-Privacy Trade-off

In our setting, usability can be measured along three dimensions: latency, monetary cost, and implementation overhead. However, we would like to stress that Preech is not designed for real-time speech transcription. Hence, latency is not a primary concern for Preech. Nevertheless, we include it in the following discussion for the sake of completeness.
Latency Evaluations: Note that all the operations of Preech are performed on speech segments. Hence, the latency is linear in the number of segments. We evaluate the end-to-end system latency per segment (with length ~ 6s) for the OSP, the CSP, and Preech; the latency values are 2.17s, 1.70s, and 14.90s, respectively. We observe that the overhead of Preech is mostly attributed to the many-to-one VC

(11s per segment on average). When voice cloning (or one-to-one VC) is applied instead, Preech's end-to-end per segment latency reduces to 3.90s (or 11.47s) at the expense of a privacy loss as discussed in Sec.3.7.

Vocabulary Size: Considering a larger $\mathcal{V}$ (Sec. 3.4) increases the scope of the DP guarantee. For example, adding external words provides protection against statistical analysis like text classification (Sec.3.7). However, larger $\mathcal{V}$ results in an increased amount of dummy segments and hence, increased monetary cost (Table 3.3). For example, extending $\mathcal{V}$ by $\sim$ 1000 out-of-domain words for the Carpenter dataset incurred a total cost of \$25 at $d = 15$.

Distance Parameter d: As explained in Sec. 3.4, larger the value of d, greater is the scope of privacy. However, the amount of required noise increases by d. For example, for the dataset VCTK p266, increasing d from 2 to 15 increases the cost by roughly \$5 (Table 3.3).

Utility-Usability Trade-off

The following control knobs provide a venue for customizing the utility-usability trade-off.

Number of CSPs: As discussed in Sec. 3.4, using multiple CSPs reduces the amount of dummy segments (and hence, the monetary cost) in Preech. However, it comes at the price of utility; the transcription accuracy of the different available CSPs varies. For example, from Table 3.1, we observe that AWS has a higher WER than Google. Thus, using multiple CSPs may result in a lower mean utility.

One-to-One VC: As discussed above, one-to-one VC technique has lower WER than many-to-one VC technique (Table 3.2). However, it requires access to representative samples of the source speaker voice for parallel training thereby limiting scalability for previously unseen speakers (Sec. 3.4).

3.8 Related Work

In this section, we provide a summary of the related work.

Privacy by Design: One class of approaches redesigns the speech recognition pipeline to be private by design. For example, Srivastava et al. proposes an encoder-decoder architecture for speech recognition [135]. Other approaches address the problem in an SMC setting by representing the basic operations of a traditional ASR system using cryptographic primitives [136]. VoiceGuard is a system that performs ASR in the trusted execution environment of a processor [137]. However, these approaches require redesigning the existing systems.

Speech Sanitization: Recent approaches have considered the problem from a similar perspective as ours. They sanitize the speech before sending it to the CSP. One such approach randomly perturbs the MFCC, pitch, tempo, and timing features of a speech before applying speech recognition [109]. Others sanitize the speaker's voice using vocal tract length normalization (VTLN) [138, 139]. A recent approach modifies the features relevant to emotions from an audio signal, makes them less sensitive through a GAN [106]. Last, adversarial attacks against speaker identification systems can provide some privacy properties. These approaches apply minimal perturbations to the speech file to mislead a speaker identification network [140, 141].

These approaches are different from ours in two ways. First, they do not consider the textual content of the speech signal. The only exception is the approach by Qian et al. [139], which addresses the problem of private publication of speech datasets. This approach requires a text transcript with the audio file, which is not the case for the speech transcription task. In addressing the textual privacy of a speech signal, Preech adds indistinguishable noise to the speech file. The proposed techniques fail to provide this property. Second, the approaches above only consider voice privacy against a limited set of features, such as speaker identification or emotion recognition. Preech applies many-to-one VC to provide perfect voice privacy.

3.9 Conclusion

In this chapter, we have proposed Preech, an end-to-end system for speech transcription that (1) protects the users' privacy along the acoustic and textual dimensions at (2) an improved performance relative to offline ASR, (3) while providing customizable utility, usability, and privacy trade-offs.

Chapter 4

Mystique: Analog Attack on Speaker Identification

4.1 Introduction

As a primary mechanism for human communication, speech is a natural vehicle for human-computer interaction (HCI). Fueled by advancements in Machine Learning (ML), everyday devices and services accept speech as input; users can seamlessly control their smart devices and communicate with automated customer services. This convenience brought the need to authenticate users when speech is the primary interaction modality. Companies deploy automatic speaker identification systems (ASI) that pack ML-based models to authenticate users based on their voiceprint [142, 143].

Building upon prior chapters, this chapter delves into vulnerabilities within speaker identification systems. We investigate the potential for impersonation attacks and vulnerabilities within existing defenses, uncovering a new, previously unexplored, attack vector. The underlying theme across these attacks is the manipulation of speech signals, with adversaries aiming to deceive the speaker identification system. This exploration builds on our examination of privacy concerns in voice-enabled technologies, providing a more comprehensive understanding of security challenges in the context of machine learning applications relying on speech as a primary modality for human-computer interaction.

Speaker identification systems are vulnerable to an array of attacks such as speech synthesis [144, 145, 146], voice conversion [147, 148, 149], replay attacks

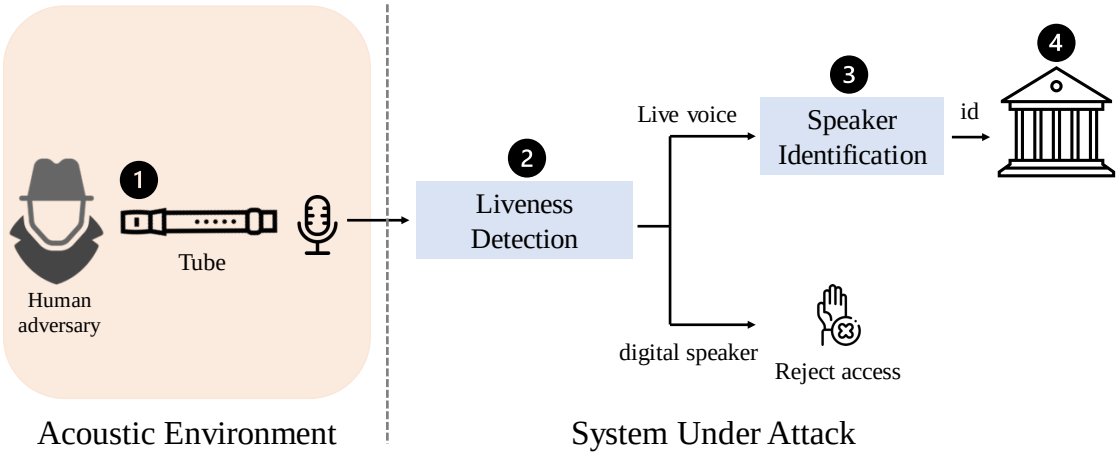


Figure 4.1: Overview of Mystique voice impersonation attack. Left: Acoustic environment fall's under the adversary's control. Right: the system under attack setup. ❶ The adversary speaks through an adversarially designed tube. ❷ A liveness detection model confirms the liveness of the captured voice. ❸ An automatic speaker identification model recognises the identity of the adversary as the target speaker. ❹ The secure system gives access to the adversary.

[150], and adversarial examples [151, 152, 153]. The adversary generates and feeds the speaker identification system a speech sample to impersonate a target speaker. While the attack techniques differ, they share a common principle: *the attacker manipulates the speech signal in the digital domain* and potentially plays it through a speaker. Note that even physical adversarial examples in the vision domain follow the same principle. Generating these examples requires obtaining a signal (such as a speech recording or a visual patch) by solving an optimization problem in the digital domain and later realizing it in the analog domain.

Current defenses leverage this observation and employ mechanisms to detect the digital attack artifacts in the input signal [154, 78, 155]. These defenses target either the (1) physical properties of the speaker *e.g.* their physical presence [156, 157] or (2) properties of the speech speakers produce *e.g.* the energy distribution of different harmonics [158, 101]. The resulting unified acoustic pipeline constrains the attacker when generating the attack samples, thus increasing the cost of the attack [159, 154, 155]. Generally speaking, the defense literature makes a basic assumption that *the attack source is not human*. In this chapter, we challenge it by asking this question: *Is it possible to attack speaker identification systems using analog manipulation of the speech signal?*

Answering this question in the affirmative has critical implications on using

ML to detect and identify human speakers. An analog transform of the speech signal to evade speaker identification challenges the *identifiability* assumption that underlies various acoustic tasks; human characteristics can no longer be uniquely identified from their speech. An attacker can control the propagation medium to affect the speaker identification task. Towards that end, we present Mystique, a *live spoof attack*, which enables analog transformations of speech signals. Mystique allows the attacker to transform their voice for inducing a targeted misclassification at the ASI system, effectively impersonating a target victim.

Realizing Mystique requires us to satisfy four conditions. First, the analog transform must occur on live speech. Second, an arbitrary speaker should be able to impersonate another arbitrary victim; *i.e.*, the attacker needs not be a professional vocalist or have any impersonation experience. Third, the transform should directly impact the ASI model prediction. Fourth, the transform can be mathematically modeled to be incorporated in the attack optimization objective. Mystique exploits the acoustic resonance phenomenon to satisfy these conditions. Acoustic resonance is a physical transform where objects vibrate to specific frequencies. Acoustic resonance allows an object to act as a frequency filter, amplifying some frequency components and dampening others.

Mystique uses hand-crafted tubes to apply the adversarial resonance transformation to the speaker's voice. We chose tubes as our attack's physical objects for two reasons. First, tubes are ubiquitous and inexpensive; they are available in hardware stores in different dimensions. Second, there is extensive literature on acoustic modeling of musical wind instruments, most of which have cylindrical or conical shapes. Note that the same methodology can be extended to arbitrary shapes using wave simulation and numerical analysis [160, 161].

To realize Mystique, we model the tube resonator as a band-pass filter (BPF) transform; the tube dimensions fully define the filter. Next, we develop a black-box optimization procedure over the filter parameters (tube dimensions) to trick the ASI model into recognizing the voice of a chosen target speaker. We apply an evolutionary algorithm (Sec. 4.4) that uses the ASI model to find the optimal tube dimensions for a given target. An adversary can use these parameters to realize a tube that would match their voice to a target speaker.

We perform extensive evaluation of Mystique on two state-of-the-art ASI models and five spoofing detection baselines. We validate Mystique on a standard speaker identification dataset, VoxCeleb, and on live speech by conducting a user study

of 14 participants. We build a physical recording setup and evaluate Mystique physically. We confirm that Mystique's adversarial tubes succeed in performing over-the-air impersonation attack in the real world.

This chapter makes the following contributions:

- We show that a human can directly produce analog audio adversarial examples in the physical domain. This adversary bypasses current acoustic defenses based on liveness and (presumably uniquely) identifying characteristics of the speaker, such as voice pitch.
- We demonstrate, using commonly available plastic tubes, that an attacker can change the properties of their speech in a systematic way and manipulate ML models. For example, an adversary can impersonate 500 other speakers using tubes. Moreover, Mystique is only 23% detectable by the best ASVspoof 2021 spoofing detection baseline that has 100% accuracy on classifying natural (*i.e.*, no tube) recordings as live.
- We run our attack on live speech to confirm its practicality. We perform a user study and show that the attack is successful over-the-air on live speech with a 61.61% success rate. We conduct a human impersonation study as a baseline and find that its success rate is only 6.2%.
- We discuss a set of strategies to detect the attack and add a discussion of limitations and future work.

4.2 Acoustics Background

In this section, we introduce background concepts on acoustics and human speech modeling.

Acoustic Resonance

Resonance is a natural phenomenon in which objects vibrate when excited with a signal that contains specific frequency components [162]. These frequency components are referred to as the resonance frequencies, and they contain the fundamental frequency f_0 (object's natural frequency) and its harmonics f_i. A resonating object acts as a *filter* that magnifies the resonance frequencies, and filters out other frequencies in the excitation signal. The resonance vibrations encounter

resistance and losses that define the filter sharpness—referred to as the quality factor Q. The filter's f_0 and Q are usually well defined by the object's shape and properties.

Acoustic resonance happens to sound waves that travel inside a hollow object, such as a tube, when it forms a standing wave [163, 162]. This phenomenon is observed in wind instruments musical notes. Similar to musical tones, human speech is produced by resonance inside the speaker's vocal structure. In Mystique, we exploit this phenomenon and our understanding of the human speech to design a physical speech filter using tubes and perform targeted attacks on ASI.

Resonance Frequency. In (cylindrical) tubes, the fundamental resonance frequency $f_0 = c_{air}/\lambda$ (Hz), where c_{air} is the speed of sound in air, and λ is the standing wave wavelength. For open-ended tubes, as in our use case, the fundamental mode $\lambda = 2L$ where L is the tube length [164]. Thus, $f_0 = c_{air}/2L$, and $c_{air} = 20.05\sqrt{T}$ (m/s) in dry air [162], where T (K) is the thermodynamic temperature. These equations, however, do not consider the tube diameter and air humidity. A more accurate equation is:

$$f_0 = \frac{c_{air}}{2(L + 0.8d)}, \tag{4.1}$$

where d is the tube diameter, and $\Delta L = 0.8d$ is an empirical term derived from measurements [165].

Quality Factor. The quality factor quantifies the acoustic losses inside the tube. There are two main sources of losses [166, 162]: radiation loss and wall loss. The radiation loss d_{rad} is the energy loss due to acoustic radiation outside the tube [162]: $d_{rad} = 2\pi A f_0^2/c_{air}^2$, where A is the tube cross-sectional area. The wall losses happen because the air speed goes down to zero at the tube internal walls, hence, it leads to energy loss. Wall losses can be quantified by this damping factor [162]: $d_{wall} = \sqrt{\mu/\rho A f_0}$, where $\mu = 1.81 * 10^{-5}$kg/ms is the air viscosity, and $\rho = 1.18$kg/m^3 is the air density. There are other losses that are either hard to quantify, or environment-dependent, or can be ignored compared to the radiation and wall losses [167]. Thus, the tube quality factor can be approximated by:

$$Q_0 = 1/(d_{rad} + d_{wall}). \tag{4.2}$$

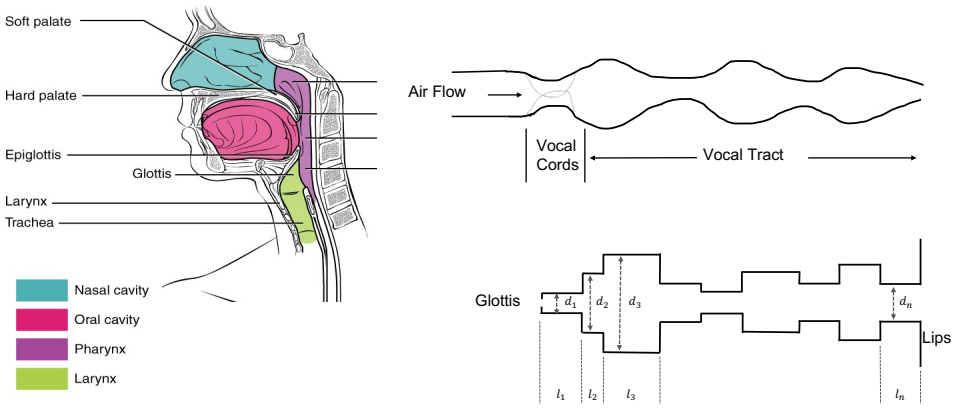
Human Speech Modeling

Biological Characteristics. Humans generate speech using three main structures [168]: the lungs, the vocal folds (glottis), and the articulators as shown in Fig. 4.2a. The lungs produce airflow and control air pressure, this airflow in turn makes the vocal folds vibrate and modulate the passing air to produce sound (audible air vibrations)—referred to as the glottal excitation. The vocal folds physical shape controls the vibrations frequency, hence, it is considered the *speech source* [168]. The vibrating air passes through the articulators—referred to as the vocal tract—such as the pharynx, the oral cavity, the tongue, the nasal cavity, and the lips. The vocal tract forms a flexible airway that shapes the sound into the final distinctive speaker voice. The moving parts, such as the tongue and lips, change their position to produce different sounds and speech phonemes. Thus, the vocal tract is considered a *linear acoustic filter* [168], and human speech production is modeled as a sound source followed by an acoustic filter.

Source-Filter Model. The glottal excitation defines the voice *pitch* and can be modeled by an impulse train in the time domain $g(t)$ and by harmonics in the frequency domain $G(f) = \mathcal{F}(g(t))$. The vocal tract can be modeled as a variable acoustic resonator $H_v(f)$ that filters the glottal excitation into speech $s(t) = \mathcal{F}^{-1}(H_v(f) \cdot G(f))$. The resonator characteristics depend on the vocal tract size and shape; *i.e.* the speaker's anatomy, and the speech phonemes vary with the tongue and lips movement. The different parts of the vocal tract are modeled as consecutive tubes [169], as shown in Fig. 4.2b. The tubes are an acoustic resonator that amplifies certain frequencies and filters out others to shape the acoustic excitation into a specific voice and speech sound.

4.3 System and Threat Models

In this chapter, we consider Automatic Speaker Identification (ASI)—a classification task that determines a speaker's identity, based on their speech [170], from a set of enrolled speakers. Typically, the identification task can be text-dependent; *i.e.* the speaker has to say a predefined utterance, or text-independent; *i.e.* the speaker can say any utterance of their choice. Text-independent ASI is more secure against



(a) Vocal Tract Structure. (b) Vocal Tract Model.

Figure 4.2: The vocal tract structure and model. (a) The structure including the glottis, the pharynx, the oral cavity, the nasal cavity, and the lips—adapted from AnatomyTool [1]. (b) Vocal tract parts modeled as consecutive tubes of different diameters.

replay attacks, and more usable as it can be embedded within other tasks such as speech recognition in a seamless interaction.

System Model. We consider a system that applies the ASI task for user identification and authentication. The system collects speech samples from its users during the enrollment phase to extract their voiceprint (speaker embeddings) and fine-tune the ASI model.

Modern ASI systems are based on speaker embedding by deep neural networks. These models capture the speaker's voice characteristics from a variable-length speech utterance $s(t)$ and map it to a vector (embedding) in a fixed-dimensional space. X-vector DNN [170, 143] is a common ASI embedding network which consists of 3 stages: (1) feature extraction, (2) speaker embedding, and (3) classification. The first stage extracts the mel-frequency cepstrum coefficients (MFCC) which reduce the dimensionality of the speech signal into a 2D temporal-spectral map, and applies voice activity detection (VAD) to filter out non-speech segments. Second, a time-delayed neural network (TDNN) maps the variable-length MFCC samples into a fixed-dimensional embedding (x-vectors) space. Finally, a softmax layer is applied on x-vectors to obtain the predicted identity of the speaker. The network is trained using a multi-class cross entropy objective.

During inference, the system collects a speech utterance from the user, and

runs the ASI task to determine the user's identity. The ASI task is the *only* access control mechanism deployed by the system. The system also applies a spoofing detection technique as a countermeasure against spoofing attacks; as we detail next in the threat model as well as Sec. 4.8.

Fig. 4.1 shows the system setup. The system runs a spoofing detector that determines whether the recorded utterance is from a live speaker or digitally produced, i.e., spoofed. If the utterance is detected to be live, the spoofing detector feeds it to the ASI model which classifies the speaker identity and grants the user access to the secure system. This system setup can be deployed for logical access applications such as phone banking services, voice assistants, and smart home devices.

Threat Model. We consider an adversary that wants to attack the ASI model to be identified as a target user. First, the adversary will not perform conventional spoofing techniques such as replay, speech synthesis, voice conversion, or digital adversarial examples to evade detection by the system's spoofing detector. Note that spoofing detection techniques (Sec. 4.8) are based on the assumption that spoofed speech is always generated by a *digital* speaker, not a live human. Instead, the adversary will *naturally* impersonate the victim's voice by changing their *live* voice using physical objects. Our work introduces a systematic reproducible technique that allows the adversary to impersonate an arbitrary speaker's voice, in the eyes of the ASI model, without using a digital speaker. The attack is analog and only allows for the use of physical objects and natural sounds.

Second, the adversary performs an audio-only interaction with the system. Hence, they have complete control over the recording environment, as shown in Fig. 4.1. They have no access to the ASI model internals; *i.e.*, a black-box attack. The adversary can only query the ASI model on inputs of their choice and get the model's output scores and label. As such, the adversary needs no recordings of the victim's speech. They only know the victim is enrolled in the ASI model. Finally, the adversary impersonates the victim in the eyes of the ASI model to gain access to their protected accounts. The attack does not target human listeners explicitly.

4.4 Attack Methodology

This section introduces our attack, Mystique, provides a theoretical intuition, and details its operation.

Overview

Fig. 4.1 displays Mystique's system and attack flow. A microphone captures the speaker's voice, validates the voice liveness, and feeds it to an ASI system. Mystique exploits the flawed assumption that spoof attacks must be generated from a digital speaker. The current ASI setup overlooks the acoustic environment attack vector. Mystique challenges these assumptions and performs an attack that is live by default. An attacker speaks through a specifically designed tube to induce a targeted misclassification at the ASI system, effectively impersonating a target victim.

Attack Description. The attack is as follows. The adversary models the tube resonator as a band-pass filter (BPF) transform (Sec. 4.4). The filter is fully defined by the tube dimensions. Next, the adversary runs an optimization function over the filter parameters (tube dimensions) to trick the ASI model into classifying the voice as a chosen target speaker. In a black-box setting, we apply an evolutionary algorithm (Sec. 4.4) that uses the ASI model score and label to find the optimal tube dimensions for a given target speaker:

$$\min_{p} \quad R(\mathsf{ASI}(s'), y_t) \quad \text{s.t.} \quad s' = F_{\text{tube}}(s, p), \tag{4.3}$$

where s is the original speech sample, p is the tube parametrization, y_t is the attack target label, R is the loss, $F_{\text{tube}}(.)$ is the mathematical model of the tube, and $\mathsf{ASI}(.)$ is the model under attack. The adversary would then purchase the required tube, and speak through it to trick the system. Therefore, the adversary is able to systematically bypass spoofing detection and attack ASI with an analog attack.

Modeling Resonance in Tubes

Modeling the filter corresponding to a particular tube is a key requirement for Mystique. We model the tube transfer function $H_{res}(f)$ as a sum of band-pass filters (BPFs), with a filter at each harmonic. The i^{th} filter $H_i(f)$ is defined by its center

frequency at the resonance harmonic f_i, and the filter width Δf_i is defined by the quality factor Q_i (Eqn. (4.5)), where $i = 1, 2, \cdots, \lfloor f_s/f_0 \rfloor$ is the harmonic number, and f_s is the speech sampling rate. The input speech signal $s_{in}(t)$ resonates at the tube's fundamental frequency f_0 and its harmonics $f_i = i \cdot f_0$. Thus, the tube output speech signal is:

$$s_{out}(t) = F_{\text{tube}}(s_{in}, p) = \mathcal{F}^{-1}(H_{res}(f) \cdot S_{in}(f)), \tag{4.4}$$

where $\mathcal{F}^{-1}$ is the inverse Fourier transform, $S_{in}(f) = \mathcal{F}(s_{in}(t))$ is the input speech spectrum, $H_{res}(f) = \sum H_i(f)$ is the tube transfer function, and $p = (L, d)$ are the tube parameters. Note that $H_{res}(f)$ is parameterized by p, but we drop this parameterization to make the notation simpler. In Mystique, we adopt a simple two-pole band filter for $H_i(f)$.

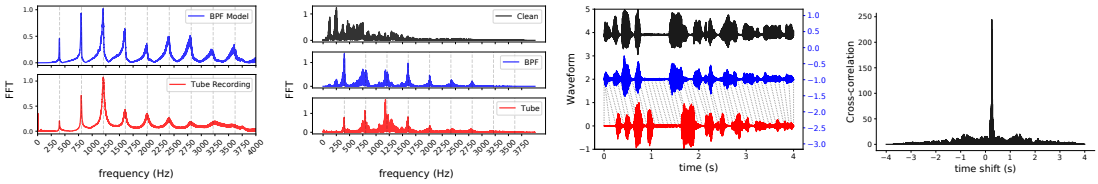
Single Tube. Given a single tube with length and diameter parameters p, Eqn. (4.1) and Eqn. (4.2) quantify the fundamental resonance parameters. The full harmonic range of f_i and Q_i are:

$$f_i = i \cdot f_0 = \frac{i \cdot c_{air}}{2(L + 0.8d)}; \quad Q_i = Q_0/\sqrt[4]{i}, \tag{4.5}$$

where i is a positive integer representing the harmonic number for open-ended tubes.

Our lab measurements revealed that there is about 1% mismatch between the theoretical (Eqn. 4.1) and measured f_0. We attribute this mismatch to the end-correction term uncertainties and air humidity. Also, we estimated Q_i empirically, as its change with f_i depends on the dominating loss for a given tube. We found that Q_i decays as $1/i$, $1/\sqrt{i}$, or $1/\sqrt[4]{i}$ give reasonable estimates and we decided to select the latter. We include both corrections in the filter formulation.

Multiple Tubes. Next, we extend the single tube model into a structure of multiple consecutive tubes of different lengths and radii to increase Mystique's degrees of freedom and the set of possible filters. The extended structure can reach a wider range of spoofed identities, hence, it increases the attack success rate as shown in Sec. 4.6.



(a) FFT of a chirp (b) FFT of speech (c) Speech waveform (d) Correlation

Figure 4.3: Resonance model validation of Tube 1 ($L = 40.6, d = 3.45$) vs its BPF model: (a) FFT of chirp, (b) FFT of a speech utterance, (c) speech wavefroms showing DTW alignment between tube and BPF signals, and (d) cross-correlation between tube and BPF waveforms.

Resonance inside connected open-ended tubes happens when the acoustic impedance between the connected tubes equal an open-end impedance [171]. This condition is mapped to the following equation for each two tubes intersection:

$$A_1 \cdot \cot(2\pi f L_1/c_{air}) = A_2 \cdot \cot(2\pi f L_2/c_{air}), \tag{4.6}$$

where A_1 and A_2 are the two tubes cross-sectional areas, L_1 and L_2 are their lengths. We solve this non-linear equation numerically to obtain the resonance frequencies f_i's.

Validation. We validate the resonance model by measuring real tubes resonance and comparing it to our BPFs model. First, we excite the tube with a 3-second chirp signal [172] that exponentially spans the frequency range from 100 to 3700 Hz. Then, we play speech samples from VoxCeleb dataset and measure the similarity between tube and BPF output signals. We use the setup in Fig. 4.5 for recording.

Fig. 4.3 shows the Fast Fourier Transform (FFT), waveform, and cross-correlation plots for a tube of $L = 40.6$, $d = 3.45$ cm. Fig. 4.3a shows the FFT of the chirp output, which is effectively the tube's transfer function $H_{res}(f)$. The vertical dotted lines indicate the theoretical resonance frequencies, f_i, which align perfectly with the measurement. Fig. 4.3c shows the waveforms with the dynamic time warping (DTW) alignment, and Fig. 4.3d shows that the waveforms are highly correlated. We also measure the DTW alignment distance for a set of 6 tubes (Table 4.1), which is a measure of similarity. The distances are 0.027, 0.03, 0.025, 0.023, and 0.021. Thus, the tube and BPF waveforms are very similar for all evaluated tubes. Therefore, the BPF model is a realistic representation of the tube resonance. The

attacker uses this model to obtain the tube parameters for a targeted attack.

Attack Intuition

Speech technology applications such as speech recognition, speaker identification, and keyword spotting are highly sensitive to the acoustic environment. Models trained on clean speech recordings often fail in real-world scenarios [173, 174, 175]. Usually, training data has to be augmented with simulated environmental effects such as noise and echo [173, 174, 175]. The same applies for speech adversarial examples. Adversarial perturbations do not succeed over-the-air when the environmental variations are not considered in the optimization objective [42, 33]. Hence, one of the fundamental intuitions behind Mystique is that if the acoustic environment falls outside the expected distribution, the model predictions will become unreliable.

Still, one can wonder why a tube (resonator) has such a high impact on the ASI model's performance. We theoretically show that tubes affect the estimated pitch. Next, we empirically validate that tube parameters are statistically significant predictors of pitch shifts between input and output signals. Such pitch shifts introduce distribution shifts w.r.t the real-world utterance datasets used to train speech models. It has been well-established that such distribution shifts reduce model performance at inference time [176, 177]. In particular, ASI is sensitive to the pitch of the speech signal; therefore, applying the tube is expected to alter the classification.

Tubes Cause Pitch Shifts

We build on the work of McAulay and Quatieri [178] who frame the pitch estimation as the solution of an unconstrained optimization of the mean square error between the Short-time Fourier transform (STFT) of a signal $s(t)$ and a sum of harmonics, parameterized by the pitch.

McAulay and Quatieri [178] use the peaks of the Short-time Fourier transform (STFT) of a time domain signal $s(t)$ to represent it as a sum of L sine waves:

$$s[n] = \sum_{\ell=1}^{L} A_\ell \exp[j(n\omega_\ell) + \theta_\ell].$$

The values of A_ℓ, ω_ℓ, and θ_ℓ represent the amplitudes, frequencies, and phases of the STFT peaks of the speech signal. Then, they find the value of ω_0 which fits $s[n]$ to $\widetilde{s}[n, \omega_0]$ as:

$$\tilde{s}[n, \omega_0] = \sum_{k=1}^{K(\omega_0)} \widetilde{A}(k\omega) \exp[j(nk\omega_0) + \phi_k],$$

where ω_0 is the signal pitch, $K(\omega_0)$ is the number of harmonics in the signal, $\widetilde{A}(k\omega)$ is the vocal tract envelope, and ϕ_k is the phase at each harmonic. Finally, the pitch is estimated by minimizing the mean squared error $\epsilon(\omega_0) = P_s - \rho(\omega_0)$, where P_s is signal's power which is a constant. Therefore, we only need to minimize $-\rho(\omega_0)$, or equivalently:

$$\max \quad \rho(\omega_0) \tag{4.7}$$

where

$$\rho(\omega_0) = \sum_{k=1}^{K(\omega_0)} \widetilde{A}(k\omega_0) \left[\sum_{\ell=1}^{L} A_\ell \left| \text{sinc}\left(\omega_\ell - k\omega_0\right) \right| - \frac{1}{2} \widetilde{A}\left(k\omega_0\right) \right]. \tag{4.8}$$

As discussed in Section 4.4, the tube results in a resonance effect, modeled as a set of bandpass filters at the resonance frequencies of the tubes. As such, some of the frequency components of $s(t)$ will be dampened. We represent this effect as $A_\ell = 0$ for $\ell \in \mathcal{L}$ as well as their submultiples $\omega_0 \in [K(\omega_0)]$, where $\mathcal{L}$ represents the set of non-resonant frequencies:

$$\begin{aligned} &\max && \rho(\omega_0) && \\ &\text{s.t.} && A_\ell = 0 && \forall \ell \in \mathcal{L}, \forall \omega_0 \in [K(\omega_0)] \end{aligned} \tag{4.9}$$

Note that Eqn. (4.9) is a constrained version of Eqn. (4.7). We can solve the latter by maximizing the Lagrangian:

$$p(\omega, \boldsymbol{\eta}) = \rho(\omega_0) - \sum_{k=1}^{K(\omega_0)} \sum_{\ell \in \mathcal{L}} \eta_{k\ell} A_\ell \tag{4.10}$$

where the matrix $\boldsymbol{\eta} = [\eta_{k\ell}]_{K(\omega_0) \times |\mathcal{L}|}$ represents the Lagrange multipliers. Instead of directly maximizing Eqn. (4.10) and finding $\boldsymbol{\eta}$, we re-write Eqn. (4.8) separating

the components in and outside of $\mathcal{L}$:

$$\rho(\omega_0) = \rho_f(\omega_0) + \sum_{k=1}^{K(\omega_0)} \widetilde{A}(k\omega_0) \sum_{\ell \in \mathcal{L}} A_\ell |\operatorname{sinc}(\omega_\ell - k\omega_0)|. \tag{4.11}$$

where

$$\rho_f(\omega_0) = \sum_{k=1}^{K(\omega_0)} \widetilde{A}(k\omega_0) \Big[\sum_{\ell \notin \mathcal{L}} A_\ell |\operatorname{sinc}(\omega_\ell - k\omega)| - \frac{1}{2}\widetilde{A}(k\omega_0) \Big], \tag{4.12}$$

is the objective function for estimating the pitch of the filtered signal. Next, substituting Eqn. (4.11) in Eqn. (4.10):

$$p(\omega, \boldsymbol{\eta}) = \rho_f(\omega_0) + \sum_{k=1}^{K(\omega_0)} \sum_{\ell \in \mathcal{L}} \Big(\widetilde{A}(k\omega_0) |\operatorname{sinc}(\omega_\ell - k\omega_0)| - \eta_{k\ell} \Big) A_\ell \tag{4.13}$$

Using the KKT conditions [179], we know for $p(\omega_0, \boldsymbol{\eta}^*)$ to be the maximizer of Eqn. (4.13), the second term should vanish. Given $A_\ell > 0$, we should have that:

$$\eta_{k\ell} = \widetilde{A}(k\omega_0) |\operatorname{sinc}(\omega_\ell - k\omega_0)|. \tag{4.14}$$

But that means $\rho_f(\omega_0) = p(\omega_0, \boldsymbol{\eta}^*)$ is the exact solution to Eqn. (4.9), *i.e.*, the equality constraint holds perfectly.

Having established that the second optimization problem is a constrained version of the first, it follows that Ω, the feasibility set of Eqn. (4.7) is a subset of Ω_f, the feasibility set of Eqn. (4.9). Then, unless $\mathcal{L} = \emptyset$ (which trivially results in $\Omega = \Omega_f$), there exists $\omega_0 \in \Omega \setminus \Omega_f$ such that ω_0 is a valid estimated pitch that has been filtered out by the tube. Therefore, we have shown that the tube will cause shifts in the estimated pitch.

Validation. We design an experiment to study the correlation between the pitch shift and the change in the classification result. We played samples from the VoxCeleb dataset through three tubes of different lengths (corresponding to different resonance frequencies). For each sample, we estimated the pitch of both signals (original and output) using CREPE [180] which provides a time-domain signal of the signal pitch. Given that the pitch varies in the duration of each utterance, we need to account for different speakers, utterances and original clip recordings to establish a generalized relationship between pitch shifts and tube parameters.

Algorithm 2 Differential Evolution

Input: s, y_t, pool size N, attack budget n, fitness function f, crossover parameter c, maximum iterations it, mutation proportion m
$A : N \times n = \text{random}(\text{pool})$
for $i = 0$ **to** it **do**
  $A_{new} : N \times n = 0.0$
  for $j = 0$ **in** N **do**
    $r_1, r_2 = \text{sample-randomly}(A)$
    $l = A_{best} + m \times (r_1 - r_2)$
    $m = c > \text{random-mask-of-size}(n)$
    $a = l * m + A_j * (1 - m)$
    if $f(a, s, y_t) > f(A_j, s, y_t)$ **then**
      $A_{new,j} = a$
    else
      $A_{new,j} = A_j$
    end if
  end for
  $A = A_{new}$
end for

We regress this pitch difference using an ordinary least squares model with a design matrix containing tube parameters and 2060 audio samples. The linear regression model achieves an $R^2 = 0.552$. Therefore, the tube parameters explain at least 55% of the pitch shift variances. P-values achieved are 1.77×10^{-26} and 2.99×10^{-149} for length and parameter, respectively, which means that these tube parameters are good regressors of the shifts introduced by the tube in a variety of recording conditions, utterances, and speakers.

Mystique's Algorithm

In Sec. 4.4, we parameterize the tubes by the quality factor Q_0 and the fundamental frequency f_0. Although, for a single-tube configuration, the search space is small enough to be brute-forced within a few minutes, we find that in many cases we can speed up the attack using optimization. More precisely, we experiment with a gradient-free non-convex optimization algorithm from a family of evolutionary algorithms called *differential evolution* (DE) [181].

Algorithm 2 describes our DE approach with *best2exp* strategy. The algorithm performs the tube parameters A search by picking three data samples from an underlying population and combining the best-performing one with the difference

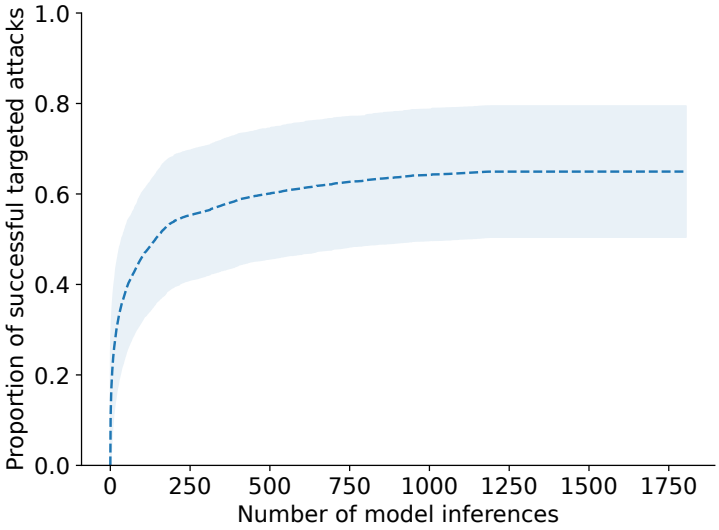


Figure 4.4: Average reachable target search performance across all of the participants with SpeechBrain model

between the other two. The algorithm is called differential since the update step includes computing the difference between a pair of samples and stochastically appending it to the third. In the search algorithm, we set boundary conditions on the tube dimensions which define the underlying population. We define the boundaries as f_0 ranges from 50 Hz to 1 kHz, and its Q_0 ranges from 5 to 100, such that f_0 falls in the typical range of human voice pitch. We sample from this range using a step size of 10 Hz for f_0 and 5 for Q_0. According to Eqn. (4.1) and Eqn. (4.2), the single tube length would range from 10 cm to 3 m, and the diameter ranges from 1 cm to 15 cm, which is a practical range. For two-tube structures, each tube length can range from 5 cm to 120 cm with a 5 cm step size, and the area ratio ranges from 1 to 10 with a step size of 1. The resultant f_i's are found from Eqn. (4.6). We set the population size $N = 100$, maximum iterations $it = 5$, and tolerance of 0.001. The attack is performed in a black-box fashion, requiring only the target class score of the ASI model. Thus, the fitness function $f(A, s, y_t)$ is the ASI model's score of the target label y_t when transformation A is applied to the user's utterance s. We find that within 100 model invocations, as is demonstrated in Fig. 4.4, we could find $46\% \pm 12$ of all possible reachable targets, whereas at 250 invocations it grows to $55\% \pm 14$. Despite the relatively low performance, our DE algorithm enables the attacker to within minutes check with a reasonable probability if a user's utterance s can be transformed to impersonate a target y_t. Further results are shown in Fig. 7.4 and 7.5 in Appendix 7.6.

4.5 Experimental Setup

We design an experimental setup, comprising speech datasets, ASI models, spoofing detection models, and a physical measurement setup to evaluate our proposed attack, Mystique. Our evaluation answers the following questions:

Q1. How well does Mystique *perform as an impersonation attack on ASI models?* We instantiate Mystique on a standard dataset, VoxCeleb, using the resonance filter model. We show that Mystique can successfully attack two ASI models. Using Mystique, each adversarial speaker successfully impersonates 500 targeted victims, on average. (Sec. 4.6)

Q2. Does Mystique*'s impersonation succeed in real-world?* We build a physical recording setup and run Mystique over-the-air on VoxCeleb (Sec. 4.6). We also conduct a user study and evaluate Mystique on live speech. We show that Mystique's attack success rate over-the-air is 61% on a standard dataset and 61.61% on live speech. We also compare Mystique against human impersonation as a baseline and find that most participants were not able to reliably impersonate a target speaker with a success rate of 6.2% on average. Finally, we show that Mystique is consistent over multiple trials.

Q3. How can a defender detect Mystique*?* We study different strategies to detect Mystique. We show that while the Mystique-generated and victim voiceprints are similar, the ASI model is less confident under Mystique. Further, we show that a human can discern samples generated from Mystique. Finally, we find that Mystique is successful against baseline spoofing detection, but not against a detector trained on Mystique's samples.

Datasets and ML Models

ASI Models. We evaluate two state-of-the-art ASI models: (1) the x-vector network [143] implemented by Shamsabadi et al. [182], and (2) the emphasized channel attention, propagation and aggregation time delay neural network (ECAPA-TDNN) [183], implemented by SpeechBrain.[1] Both models were trained on VoxCeleb dataset [142, 184, 185], a benchmark dataset for ASI. The x-vector network

[1]SpeechBrain (`https://github.com/speechbrain/speechbrain/`) is an open-source state-of-the-art toolkit on Hugging Face

is trained on 250 speakers using 8 kHz sampling rate. ECAPA-TDNN is trained on 7205 speakers using 16 kHz sampling rate. Both models report a test accuracy within 98-99%.

Evaluation Dataset. Both ASI models are trained on VoxCeleb. Thus, we use VoxCeleb as our test dataset. We select a subset of 91 speakers, 45 female and 46 male speakers, that are common in the training dataset of both models. We select 20 random utterances per speaker on which both models achieve 100% accuracy.

Spoofing Detection Models. We evaluate two spoofing detection techniques, (1) ASVspoof baselines and (2) Void. We consider two state-of-the-art baselines from the ASVspoof 2021 challenge[2] for physical access (PA) and logical access (LA) tasks. The PA task objective is to discriminate between live human speech and replayed recordings via loudspeakers, while the LA task objective is to differentiate between live speech and artificially generated speech using text-to-speech, voice conversion, or hybrid algorithms. The LA task considers only logical attacks; *i.e.* the adversary feeds the spoofed utterance digitally to the ASI model and does not play it over-the-air. Thus, the PA and LA tasks are designed to distinguish two different features of spoofed speech: loudspeaker artifacts, and synthetic speech artifacts. We use the official implementation[3] employing the light CNN (LCNN) model [186]. However, each is trained on a task-specific dataset from ASVspoof 2019 challenge: *bonafide* and *replayed* samples for the PA-LCNN model, and *bonafide* and *synthetic* samples for the LA-LCNN model. The second spoofing detection technique is Void (Voice Liveness Detection) [45], a recent high-performing system that uses spectral analysis to detect synthetic speech. It extracts 97 spectral features to train an SVM model. The key assumption is that live speech power is higher at low frequencies than at high frequencies, while synthetic speech power is linearly spread out across the frequency range. This makes Void a good candidate for detecting Mystique since the resonance effect redistributes the speech power and amplifies the power at f_0 and its harmonics f_i as shown in Fig. 4.3 and Sec. 4.4. We use Wenger et al.'s implementation [187], where they train three models on the ASVspoof dataset: (1) SVM, (2) Light CNN [188], and (3) a custom 5-layer CNN.

[2] `https://www.asvspoof.org`

[3] `https://github.com/asvspoof-challenge/2021`

Live Human Impersonation. We conduct a user study to test Mystique on live speech, involving three stages.

– In the first stage, each participant records the first 50 utterances of the arctic dataset[4] using a microphone, without a tube. Since ASI is a text-independent task, we did not place any requirements or assumptions on the utterances' linguistic content. The use of the arctic dataset is an arbitrary choice. We then apply Mystique on these recordings to impersonate victims enrolled in the ASI models—speakers from VoxCeleb.

– In the second stage, we validate Mystique's success rate by conducting the attack over-the-air. We select three representative tubes that are common between the impersonation attacks of all participants. We ask each participant to speak each utterance through each tube and compare the live classification result to the one obtained from the filter. We ask the participants to maintain the same speaking style and not to press their lips against the tube opening as it creates non-linear transformations not captured by Mystique's model.

– In the third stage, we ask the participants to impersonate from 1 to 8 target speakers, based on their capacity. We select the targets from the successful impersonations using Mystique. Each participant watches videos of the target (celebrity) speaker till they feel confident about impersonating them, which took from 5 to 20 minutes each. Then, the participant is allowed five attempts to impersonate the target using their own words; i.e. they were not given a specific script to read.

We recruited 14 individuals[5] (7 males, 7 females, age:18-30). We obtained IRB approval from our institution to conduct the study. We collected no personal information, obtained informed consent from each participant, and followed health protocols. We use the ASI models described above, without retraining as to mimic a realistic attacker, which would attack black-box models. We use the physical setup, described below, to conduct the user study.

Physical Setup for the Attack

We design and implement a measurement setup to conduct the attack over the air. Fig. 4.5 visualizes our setup which comprises tube(s), a recording device, and the recording environment.

[4] `http://www.festvox.org/cmu_arctic/`

[5] Two participants abstained from conducting the third stage of the study.

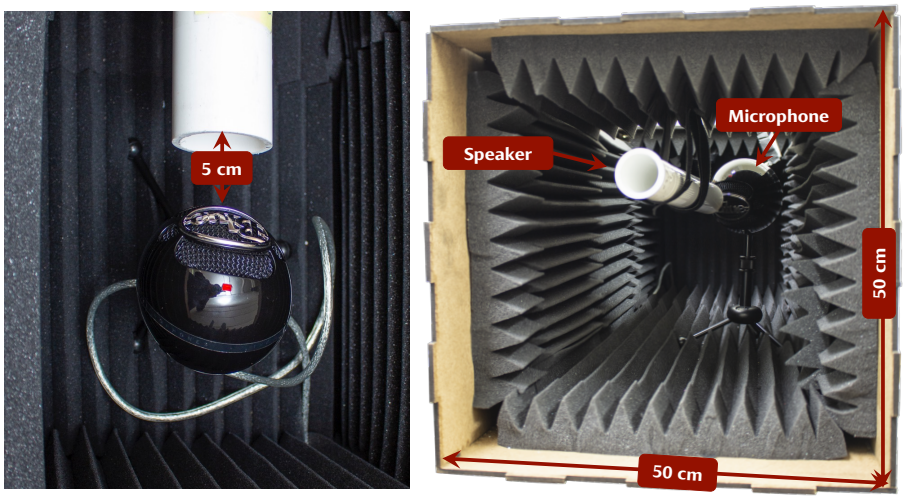


Figure 4.5: The recording setup: top view (left) and front view (right).

Tubes. We use two sets of tubes in this work. We conducted the single-tube experiments using six PolyVinyl Chloride (PVC) pipes purchased from a hardware store. Their dimensions are listed in Table 4.1. For the two-tube structures, we 3D printed the tubes for a fine-grained control over the tubes radii which impacts the resonance frequency (Eqn. 4.6). We used Formlab's Form 2[6] printer and Black Resin[7] material. We print the tubes with a 50 μm resolution for a smoother finish and a thickness of 2 mm, no support material was on the inside of the tube. The tubes are connected using High Density Fiberboard (HDF) rings at run time, for tube reusability, as shown in Fig. 7.11 in the Appendix. We constructed three two-tube structures whose dimensions are in Table 4.2. For both sets, we select the tubes dimensions based on our observations from Sec. 4.6 experiment.

Recording Environment. We conducted the experiment in a $8 \times 3.6 \times 3.6$ m lab space. We built an audio chamber to prevent interference of the tube's input and output sounds, and isolate the experiment from the background noise and speech interference from adjacent rooms; this helps unify the acoustic environment throughout the experiments. The chamber is a wooden box lined with acoustic panels to absorb the noise and minimize reverberation. We attached floating suspension loops to the chamber's ceiling to hold the tube in the air as shown in Fig. 4.5. Suspending the tube minimizes its surface mechanical vibrations. We used a Blue

[6] https://formlabs.com/3d-printers/form-2/
[7] https://formlabs.com/store/black-resin/

snowball microphone,[8] placed as Fig. 4.5, to capture the tube output signal. The setup is inspired by the design of musical instrument measurement environments. We use a Google Pixel 2 phone as a digital speaker to play sound over-the-air. A MacBook Pro laptop controls the recording. We used `python-soundevice` library to automate the recordings[9].

4.6 Mystique **Evaluation**

We conduct the following experiments to answer the three questions from Sec. 4.5 in detail.

Impersonation Attack at Scale

First, we test Mystique's impersonation attack feasibility on the full test set to address the first evaluation question. We run Mystique on the VoxCeleb (91 speakers) test set, representing the adversarial speakers, and find the range of successful impersonation attacks and the corresponding set of adversarial tubes. In this experiment, we consider structures of N-tubes, where $N \leqslant 2$. Hence, the resonating frequencies depend on three parameters (degrees of freedom): the tubes length L_1, L_2, and the tubes cross-sectional area ratio: $\mathtt{ratio}_A = (d_2/d_1)^2$.

For each adversarial speaker, Mystique attempts to impersonate every enrolled speaker in the ASI model; 7205 in SpeechBrain and 250 in X-Vector. Mystique searches for the BPF filters parameters that trick the ASI into identifying the adversarial utterance $F_{\text{tube}}(s, p)$ as the target victim speaker, y_t, using the DE algorithm 2.

Fig. 4.6 shows the number of target IDs from SpeechBrain that an attacker could impersonate using Mystique *theoretically*; i.e., the number of target IDs where Mystique successfully finds a tube configuration that fools the model for each attacker. Since real-world requirements constrain the search and Mystique has little degrees of freedom, the algorithm might not find a tube for each source-target pair. Fig. 7.9, in the Appendix, shows the same for the x-vector model. As the figure shows, by optimizing the tube dimensions, Mystique can successfully impersonate a wide range of victim speakers. Specifically, a speaker can impersonate 500 (out of 7205) target speakers on average on SpeechBrain model and 137 (out of 250)

[8] `https://www.bluemic.com/en-us/products/snowball/`

[9] `https://python-sounddevice.readthedocs.io/en/0.4.4/`

on the x-vector model. Recall that the models are initially 100% accurate on the selected evaluation dataset. Hence, this experiment shows that Mystique is capable of forming an adversarial impersonation attack on ASI models. Next, we analyze the adversarial tube (BPF) parameters and the demographic distribution of the predictions to interpret how the attack works. We report three findings.

First, the attack is most effective when f_0 lies in the frequency range $f_0 \leqslant 400$ Hz with a high quality factor $Q_0 \geqslant 50$ as shown in Fig. 7.7. This observation matches our intuitions from Sec. 4.4; the significant f_0 range falls within the typical human pitch range. An adult woman's pitch range is 165 to 260 Hz on average, and an adult man's is 85 to 155 Hz. Moreover, the low frequency speech range carries more information than the higher frequency range [49]. Hence, this range of f_0 will have a stronger impact on the pitch, the significant spectrum, and the model prediction. Also, a high quality factor means a sharper filter; and fine-grained selection.

Second, Mystique is 80% more successful on impersonating same-sex targets than cross-sex. Fig. 7.8, in the Appendix, shows the prediction confusion matrix split by the attacker-victim speakers' sex. The figures show that the cross-sex speakers' submatrix is sparser than that of the same sex.

Third, we find that Mystique impersonates different victims when optimizing for different utterances of the same speaker (attacker). Hence, the attack is not utterance (text) independent. We attribute this observation to two reasons: (1) ASI models are not perfect in separating the linguistic content and voice biometrics; the model prediction varies with the spoken utterance, (2) the attack's pitch shift and voice transformation is the result of Mystique's transformation applied on the spoken utterance original spectral content.

Over-the-air Attack

We validate Mystique's impersonation attack over-the-air using our physical setup in Fig. 4.5 to answer the second evaluation question. We conduct this experiment on VoxCeleb as a standard dataset for ASI—Sec. 4.6, and also on live speech from our user study participants—Sec. 4.6.

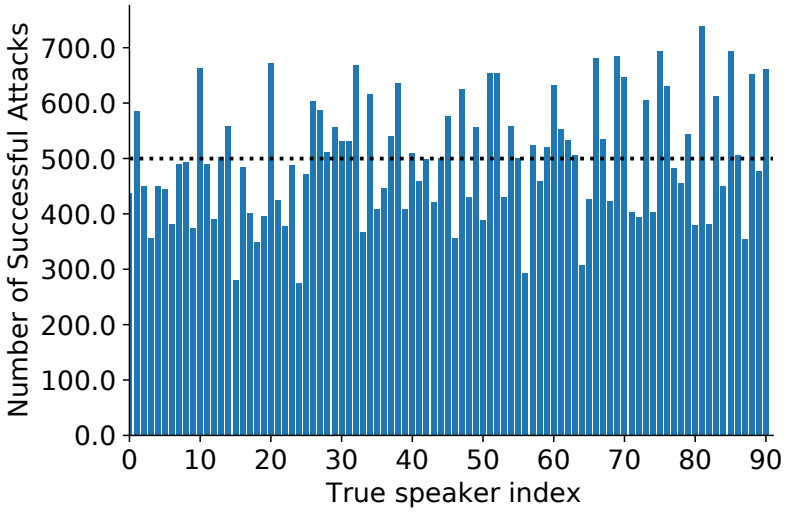


Figure 4.6: Successful impersonation attacks (out of 7205) on SpeechBrain model for each adversarial speaker from VoxCeleb. The dotted line shows the average number of successful attacks per speaker.

Tube	Tube Dimensions		Resonance Parameters		X-Vector False Predictions			SpeechBrain False Predictions		
	L (cm)	d (cm)	f_0 (Hz)	Q_0	Real	Filter	Match	Real	Filter	Match
1	40.6	3.45	402.16	58	158	141	64 (45.40%)	158	238	111 (46.64%)
2	61.3	4	270.70	68	123	194	75 (38.66%)	134	255	106 (41.57%)
3	87	5.2	191.48	77	202	242	101 (41.74%)	198	308	141 (45.8%)
4	99.4	3.45	170.89	64	325	174	77 (44.25%)	220	200	121 (60.5%)
5	120.3	5.2	140.20	79	190	167	95 (56.89%)	210	351	146 (41.6%)
6	154	5.2	110.36	76	176	108	63 (58.34%)	179	185	114 (61.62%)

Table 4.1: Evaluation of Mystique over-the-air for 40 speakers $\times$ 20 utterances: 800 total inferences. *Real*: # successful attacks of the real tube, *Filter*: # successful attacks of the corresponding filter model, *Match*: the number (percentage) of matched attacks between the filter and real tube.

Standard Dataset Evaluation

Because of the physical resources (mainly run-time) limitations, we select a subset of the evaluation speakers to form the adversarial speakers set. We also select a subset of the possible tube dimensions to run the over-the-air attack. Specifically, we randomly select 40 speakers, 20 males and 20 females, out of the 91 speakers dataset. There are 20 utterances for each speaker; a total of 800 four-second long utterances. The subset is balanced and representative of the full dataset. For the single-tube setting, we select 6 random tubes of various dimensions, listed in Table 4.1, which have f_0, Q_0 in the most significant range—Fig. 7.7 in the Appendix.

Tube	Tube Parameters (cm)				f_0 (Hz)	Attack Success Rate
	L_1	d_1	L_2	d_2		
7	9.53	2.1	10	1	853.1	66.6%
8	11.44	0.98	8.9	3.4	901.55	50%
9	14.53	2.1	10	1	600.4	100%

Table 4.2: Two-tube structures f_0 and attack success rate over-the-air.

We purchase them from the hardware store. While for the two-tube setting, we build three structures of 3D printed tubes as described in Sec. 4.5; their parameters are listed in Table 4.2.

We use the Pixel phone to simulate the speaker and play the VoxCeleb utterances over-the-air for all tubes. We record the tube output sound using the physical setup. We place the speaker on a separate tripod to allow acoustic propagation only through the air; *i.e.*, no sound is transmitted to the microphone via vibrations through the recording table. We allow a 3-second silence between consecutive utterances. We repeat the recordings 6 times to account for any environmental variations and to evaluate the attack reliability and consistency.

Single-Tube. Table 4.1 shows the number of successful attacks (impersonated targets) per tube and compares it to the successful attacks using the filter model. First, "Real" columns (6 and 9) report the number of successful attacks of the 40 speakers using the real tubes. Each speaker can impersonate up to 5 speakers identities on average using an individual tube, depending on the attacker's spoken utterance. As discussed in Sec. 4.6, we found that different utterances sometimes lead to different impersonated victims per attacker-tube pair. Second, the "Filter" columns (7 and 10) show the number of successful attacks using each tube's BPF model. The filter's successful attacks are of the same magnitude as the real tube. Finally, the "Match" columns (8 and 11) show the matching rate between the real and simulated tubes attacked identities. The match rate ranges from 38.7% to 61.62%, 48% on average. Hence, Table 4.1 confirms that speaking through a tube forms a real and effective attack on the ASI task, and the linear BPF model (Eqn. 4.4, 4.5) is a reasonable approximation of the resonance effect. A more accurate model is to use wave simulation engines at the expense of increased computation complexity.

Finally, we assess the attack's reliability over multiple trials. We measure the model's prediction consistency rate—defined as the percentage of consistent predictions across six runs. Table 7.4, in the Appendix, shows the consistency rate per tube, on average 84% of the predictions are consistent over six runs.

Two-Tube. Similarly, Table 4.2 shows Mystique's performance over-the-air using the two-tube configurations. The success rate is the percentage of matched successful attacks between Filter and Real tubes impersonated identities. Mystique's targeted attack succeeds more than 50% of the time.

Live Impersonation Attack

We run Mystique on 14 participants' natural recordings, 50 utterances each, and find the set of theoretically successful attacks (impersonated identities) per participant. Fig. 4.7 shows the number of successful attacks on the SpeechBrain model. Fig. 7.10, in the Appendix, shows the same for the x-vector model. An arbitrary speaker can impersonate 163 (117 for x-vector) target identities on average using a single-tube.

Next, we ask the participants to speak the same 50 utterances through three of our tubes. We evaluate the recordings on the ASI models and compare them to the BPF predictions. Table 4.3 reports the percentage of Mystique's BPF impersonation attacks that also succeeded over-the-air in the live recording of each participant. The average success rate ranges from 34.84% to 78.25%, showing that Mystique reliably launches over-the-air attacks. This result is significant—live human speech varies between recording sessions, unlike *e.g.* VoxCeleb experiment with fixed recordings.

Moreover, we explore Mystique's personalization by fine-tuning the filter parameters to each participant's voice characteristics. Applying a voice envelope calibration to the filter gain increases Mystique's success rate, for most participants, up to 10%. However, it drops for a few participants, as shown in column 6 of Table 4.3. Thus, personalization is one way to further optimize Mystique, which we leave to future work. Additionally, we observe the same skew in the speaker's sex for successful attacks as in VoxCeleb (Fig. 7.8), where the cross-sex submatrix is sparse.

Finally, we evaluate the participants' impersonation capabilities without using any tubes as a baseline for Mystique's performance. The last column in Table 4.3

ID	Gender	Tube3	Tube4	Tube6	Avg	Avg_{Cal}	Human
0	F	50.0	50.0	66.67	55.56	**65.0**	0/5
1	M	58.82	81.82	57.14	65.93	43.02	0/40
2	M	66.67	72.73	77.78	72.40	**72.58**	0/20
3	F	63.64	83.33	75.0	73.99	**78.7**	0/20
4	F	66.67	58.33	71.43	65.48	**73.15**	0/20
5	M	50.0	42.86	55.56	49.47	42.29	1/20 (5%)
6	M	46.15	54.55	80.0	60.23	**60.71**	6/25 (24%)
7	F	66.67	77.78	80.0	74.81	62.22	–
8	M	43.75	42.86	54.55	47.05	**52.06**	0/10
9	M	50.0	60.0	50.0	53.33	41.6	1/10 (10%)
10	F	66.67	62.5	80.0	69.72	**75.0**	5/20 (25%)
11	M	50.0	61.54	72.73	61.42	**69.17**	–
12	M	10.0	54.55	40.0	34.84	**35.56**	0/15
13	F	80	71.42	83.33	78.25	69.44	1/20 (5%)

Table 4.3: User study participants percentage (%) of successful over-the-air impersonation attacks with and without Mystique. **bold** values are enhanced by personalized calibration.

shows the number of times the participant was able to impersonate a target by the total number of trials, where for each target the participant performs 5 impersonation trials. Note that some participants were not willing to impersonate more than one target, thus the total number of trials is not the same for all of them. This study shows that most participants were not able to reliably impersonate a target speaker, where the average success rate is only 6.22%. Specifically, 7 participants did not succeed in any trials, 3 participants were able to impersonate one target one time and failed at the 4 other attempts for the same target, and only 2 participants could succeed more than once. We noticed they could capture the accent and pitch of the target. Participant 6 impersonated 3 (out of 5) targets for (3, 2, 1) trials for the same target, while Participant 10 successfully impersonated 2 targets for (2, 3) times. Yet, Mystique significantly outperforms the strongest baseline; it impersonates 100+ victims with a success rate of up to 78.7%.

Mystique**'s Robustness**

We study different strategies to detect samples from Mystique, which include: comparing prediction confidence, human-based analysis, and state-of-the-art spoofing detection.

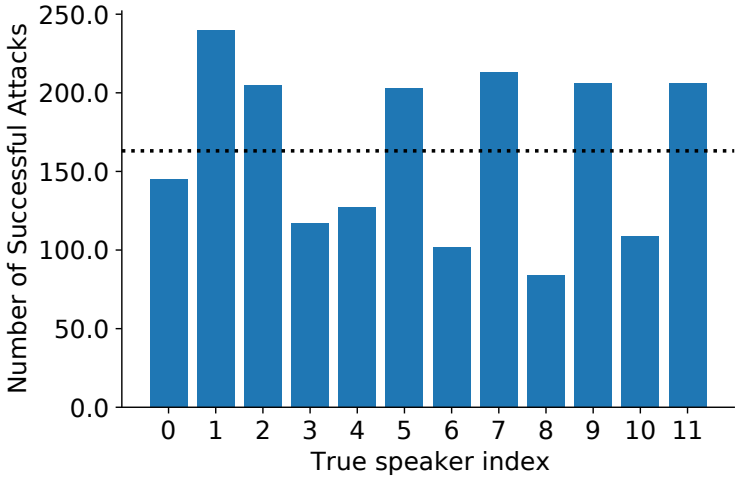


Figure 4.7: Number of successful attacks of the study participants recordings on SpeechBrain. The dotted line shows the average number per true speaker.

What is the ASI model confidence on Mystique**?**

The ASI model outputs the class (speaker id) with the highest prediction score. Here, we analyze the model's confidence in the predicted class, using the softmax score as a proxy.

Fig. 4.8 shows the distribution of the confidence scores of the model's top two classes, in the case of clean (benign) and Mystique (adversarial) samples. The figure shows that the model is less confident of its top-1 class prediction on Mystique's samples; i.e., the gap between the top-2 scores decreases. This finding arises from Mystique's samples being out-of-distribution (OOD) samples with respect to the model's training data. Hence, this analysis suggests that Mystique's threat can be weakened if the ASI model is trained to reject samples of which it is not highly confident [189].

How similar is Mystique **to the victim's voice?**

Our second detection strategy assesses Mystique's spoofed speech similarity to the victim's speech. We analyze the attack-victim speech pairs that are successful over-the-air. Specifically, we visualize the attack in the problem space (audio) and evaluate the similarity in the embedding space.

Fig. 7.6 in Appendix 7.7 shows a sample of the attacker and victim waveforms and spectral content. The samples are not visually similar and do not exhibit

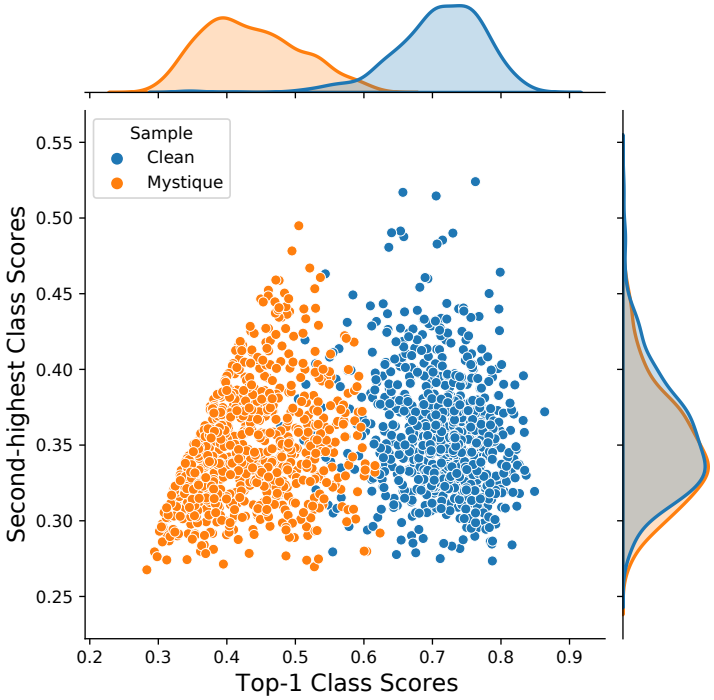


Figure 4.8: The distribution of BrainSpeech softmax scores for the top two classes on VoxCeleb clean and adversarial samples.

high cross-correlation. We attribute this result to speech being composed of two entangled characteristics: the speaker's voice and the linguistic content [149, 34]. The attack-victim samples are of different linguistic content. Note that VoxCeleb is composed of Youtube recordings of celebrities, and there is no transcript or one-to-one mapping of the linguistic content among different speakers. However, the ASI models learn a representation of the speech utterances that are presumably based on voice characteristics and invariant to the linguistic content.

Next, we analyze the ASI model embedding space similarity of the attack-victim pairs. Table 4.4 shows that the embeddings of attack and victim sample pairs exhibit high cosine similarity, compared with randomly selected non-victim speakers. Thus, regardless of the linguistic content, Mystique applies a transformation on the speech utterance that maps its embedding (voiceprint) towards the victim's voiceprint.

Cosine Similarity	Tubes					
	1	2	3	4	5	6
Attack-Victim	0.39	0.38	0.40	0.43	0.38	0.34
Attack-Non victim	0.07	0.07	0.08	0.08	0.08	0.08

Table 4.4: Average cosine similarity score of the embeddings of Mystique's successful attack utterances and its victim speakers' utterances, compared to non-victim speakers similarity scores.

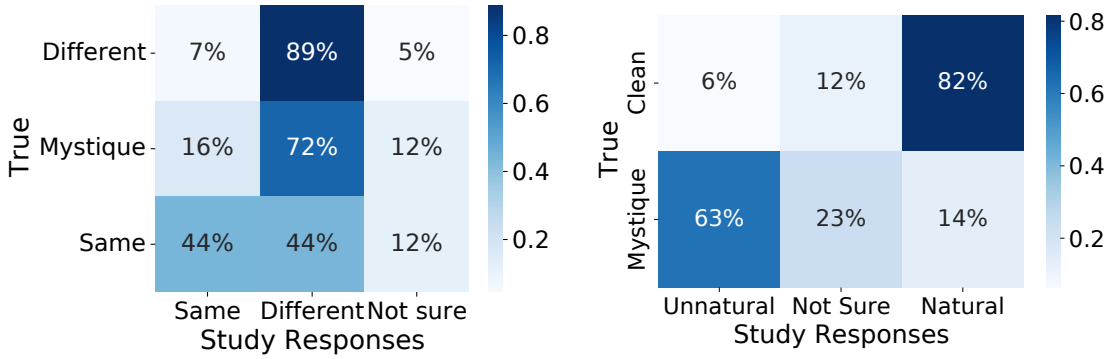
Does Mystique **confuse humans as well?**

Here, we investigate the similarity from a human point of view. Although Mystique is designed to attack ASI ML models by physically manipulating the spectral content of speech, we are curious whether it also confuses humans. To answer this question, we recruit participants to listen to two audio recordings and decide whether they belong to the same speaker and also rate the audio quality as natural or unnatural. The study is approved by IRB and is conducted on the Prolific platform.

Study design. We recruited 151 participants, each compensated $1.4 for their effort, with an average completion time of 6 minutes.

Each participant listens to 10 pairs of audio recordings from VoxCeleb; 3 pairs from each of the following cases: (a) the two recordings are clean and belong to the same speaker, (b) the two recordings are clean and belong to two different speakers of the same sex, and (c) one recording is generated by Mystique (attacker using a tube), while the other recording is of the corresponding victim's voice. The tenth pair is an attention check with two identical clean recordings. For each pair of recordings, we ask the participants two questions: (1) "*do they belong to the same speaker?*," and (2) "*how natural does the recording sound?*" on a 3-point Likert scale. We discard any responses that did not answer "same speaker" for the attention checker.

Results. Fig. 4.9 shows the distribution of responses. Fig. 4.9a shows that Mystique generated successful attacks on humans' perception 16% of the time, and was able to confuse them 12% of the time. This result is interesting given that Mystique is not optimized to trick humans. The study also shows that the participants could distinguish different speakers' voices with a high probability (89%). However,



(a) Similarity

(b) Quality

Figure 4.9: The confusion matrix of the user study responses on the audio recording similarity and quality evaluation.

Model	**Pre-trained**		**VoxCeleb**		**VoxCeleb**$_{3\,tubes}$	
	EER	FAR_0	EER	FAR_0	EER	FAR_0
LA-LCNN	29.59	76.67	6.25	19.21	8.54	22.30
PA-LCNN	31.33	98.9	7.87	33.94	14.54	67.81
Void-SVM	62.12	97.70	36.70	91.64	48.3	96.12
Void-DNN	35.39	91.88	26.12	86.61	39.1	92.55
Void-LCNN	32.91	93	27.03	91	27.64	92.55

Table 4.5: Evaluation of five spoofing detectors on the user study recordings for: (1) pre-trained detectors, (2) fine-tuned on VoxCeleb clean and tube recordings, (3) fine-tuned on VoxCeleb **without** the three tubes used in the user study. Note: FAR_0 = FAR at FRR=0%.

they were confused about the "Same" speaker recordings. Here, 44% were labeled as the same (different) speaker; *i.e.* not significantly better than random guessing. This result, supported by previous studies [190, 191], confirms that voice identity perception can be challenging for humans, especially for unfamiliar speakers. Finally, Fig. 4.9b shows that 63% of Mystique's samples sound unnatural to the participants, yet 14% sound natural. These results show how ML models perceive speech differently than humans, creating the gap that Mystique and other attacks exploit.

Spoofing Detection

Finally, we assess Mystique's robustness against a set of five representative defenses and liveness detectors as detailed in Sec. 4.5. For the five evaluated models, we report the equal error rate (EER) which is the model's error rate when the false acceptance rate (FAR) and false rejection rate (FRR) are equal. A lower EER means a more accurate detector. We also report the FAR at 0% FRR, which sets the detector's operating threshold to correctly classify all live samples.

Table 4.5 reports the detectors performance on the recordings of 12 participants from the user study (Sec. 4.6). We label the no-tube speech as *bonafide* (live) and the tube recordings as *spoofed*. The "Pretrained" column reports the performance of the pre-trained models on ASVspoof dataset. All evaluated models are unable to distinguish the tube and no-tube samples. The best performing model (LA-LCNN) labels 76% of the tube samples as bonafide. Note that these models perform very well on the synthetic data from their original papers. For example, LA-LCNN's EER = 9.26% on the ASVspoof test set [192] and Void's EER $< 12\%$ on the curated synthetic dataset by Ahmed et al. [45] and Wenger et al. [187]. Yet, spoofing detectors do not generalize to Mystique's samples.

Next, we retrain the models on our VoxCeleb clean and tube recordings as the bonafide and spoofed samples respectively. The goal is to introduce Mystique's resonance effect to the model and label it as spoofed. The "VoxCeleb" column in Table 4.5 shows that the EER and FAR_0 dropped for all models, especially LA-LCNN with only 19% of the tube samples labeled as bonafide. Then, we retrain the models again on VoxCeleb but without the recordings from the tubes (3, 4, 6) that we use in the user study. The last column "$VoxCeleb_{3tubes}$" EER and FAR_0 values increase by 2% to 85% relative to the VoxCeleb column. When the exact tubes used by the participants in the test set are not part of the training data, the performance drops significantly. Thus these models overfit to their training distributions. Previous work [193, 194, 195] has reported the same observation; spoofing detectors hardly generalize to unseen transformations in the training data, which questions the security of voice-based authentication.

4.7 Discussion

Defenses. Having established a major vulnerability in spoofing detection systems leads to a question on how one stops such attacks. We show that defenses trained on samples of the resonance transform can detect Mystique and limit its effectiveness. However, it is not clear whether such a defense approach is reliable, or even desirable. An attacker can simply use objects with different filter profile to render the defense unsuccessful; the defender cannot predict what filter the attacker would deploy. A better defense would have to incorporate properties of the medium and other modalities to rely on multiple factors, not just the speakers features.

Reproducibility. From formulating the original idea to completing the experiments, this chapter took around a year. We make a note of the things that slowed us down significantly and required non-trivial debugging. First, the use of Bluetooth or Wifi operated devices introduces significant problems because of occasional variable lag and interference. Second, during the theoretical and practical matching, it is important to isolate the setup as much as possible. In our case, matching f_0 and Q without the acoustic chamber was extremely challenging. Third, distance to the microphone and its' directionality matters—nothing should be blocking the opening of the tube, as otherwise it leads to additional echo and changes the filter as reported in the measurements literature [165]. Fourth, experiments ran on different days lead to different results, because of a change in speed of sound with temperature and humidity – its best to conduct hardware calibration and the evaluation on the same day. Finally, when producing tubes with a 3D printer, the material on the inside of the tube should be smooth.

Limitations. Despite highlighting a flaw in the current defenses design, there are a number of limitations in the current evaluation. First, we only considered simple tube structures, restricting the range of possible adversarial transformations. Second, we run the attack in a static recording environment, limiting its deployment in more practical situations where the adversary can be visually observed or has partial control over the acoustic environment or experiences acoustic effects such as noise and interference. Third, we evaluated a small number of speakers and utterances, potentially underrating the overall attack performance. Fourth, the resonance effect sounds unnatural to humans, other transforms should be explored

to have a more subtle impression on human listeners. Finally, Mystique needs access to the ASI model scores to perform the DE algorithm. However, Mystique can omit this requirement and perform an exhaustive search over all possible tube parameters, at the expense of the time complexity.

Future Work. We provide some directions to address Mystique's limitations. First, physical effects such as natural sounds and acoustic meta-materials should be explored to provide higher degrees of freedom and a less susceptible attack. Second, our evaluations suggest that the linguistic content can be optimized per each attacker-victim pair. Moreover, Table 4.3 suggests that attack personalization can boost its success rate. Third, Mystique can be made model-independent by performing the optimization on the estimated pitch as a proxy of the ASI model's decision as explained in Sec. 4.4.

4.8 Related Work

The literature on computer-based voice authentication is vast, and dates back to at least 1960s [158].

Attacks on ASI. We start by describing the four most common attacks: (1) speech synthesis, (2) voice conversion, (3) replay attacks and (4) adversarial examples. In *speech synthesis*, an adversary trains a speech synthesis model on samples recorded from the victim speaker. The adversary uses this model to convert text into speech in the victim's voice [144, 145, 146]. Alternatively, voice conversion converts spoken utterances into the victim's voice [147, 148, 149]. In *replay* attacks, the adversary records the speaker's voice and replays the recorded speech [150]. Finally, many modern ML-based ASI models inherit the vulnerability to adversarial examples using standard gradient-based attacks [151, 152, 153].

Defenses against Acoustic Attacks. What these attacks have in common is that the adversarially-generated sample would need to be generated, and transmitted digitally and reproduced through a (digital) speaker. Defense mechanisms, therefore, include (1) detecting the electronic footprint of the digital speaker (known as spoofing detection), or (2) verifying that the speaker is a live human.

Spoofing detection relies on patterns extracted from the acoustic signal to classify it as a legitimate or fake sample. Chen et al. [101] used a smartphone's magnetometer to detect the use of a loudspeaker. Blue et al. [196] tell electronic and human speakers apart by analyzing individual frequency components of a given speech sample. Yan et al. [197] calibrated individual speakers in the near field of the speakers to tell humans and electronic speakers apart.

Second, liveness detection leverages other sensing modalities such as visual, acoustic and EM signals to determine the liveness of the acoustic signal. Meng et al. [159] used an active radar to project a wave onto the face of the speaker and then detect shifts introduced to it from facial movement. Zhang et al. [157] analyzed hand movement to detect live speech by turning a smartphone into an active sonar.

Finally, there exists a class of defenses that restrict the attack surface by reducing attacker capabilities. Zhang et al. [198] used individual recordings from a stereo microphone to calculate time difference of arrival to detect replay attacks. Blue et al. [199] used two microphones to restrict the adversary to a 30 degree cone and protect against hidden and replay commands. Wang et al. [78] used correlates from a motion sensor to detect and reject hidden voice commands.

Physical Adversarial Examples. *Physical* adversarial examples are common in the vision domain, but have not been produced for acoustic tasks. Example adversarial objects include eyeware [200, 201], tshirts [202, 203], headwear [204, 205] and patches [206]. Although these objects were recreated in the real world, there is an important distinction here. These objects all apply perturbations that were initially designed for the digital space and then retrofitted with sophisticated machinery such at printers to realize them in the physical domain. Our attacks, on the other hand, directly restrict the search space of perturbations to those that can be easily realized physically. Most importantly, our attacks target a different property of the physical world—we use the environment to shape the signal, rather than exploit errors in the ML model.

4.9 Conclusion

We demonstrate that a human adversary can reliably manipulate voice-based identification systems using physical tubes, *without access to the victim's speech*. Our attacks highlight acoustic intricacies that were largely ignored by prior literature,

namely, the acoustic environment. Current defenses assume that the adversary is non-human and focus on verifying this assumption. Our human-produced attacks show that this assumption does not hold in the first place. In this chapter, we demonstrate that the subjective nature of speech can be exploited to jeopardize the security of a critical system. Concretely, a fundamental question to consider in speaker identification is whether a person's identity can be accurately established despite the transformation of their voice.

Chapter 5

Semantic Robustness and Fairness

5.1 Introduction

As our exploration of privacy and security in machine learning unfolds, this chapter shifts our exploration from audio to visual technologies, focusing on face recognition and its security implications. This chapter bridges the gap between voice identification systems and visual authentication, highlighting unique challenges in facial data processing. We will examine the technical intricacies of face recognition technology and explore the role of generative AI in creating facial images. The aim is to provide a clear understanding of how privacy and security considerations uniquely manifest in the realm of visual ML applications, drawing parallels and contrasts with the auditory systems discussed earlier.

Automated face recognition technology has rapidly expanded across various industries, including commercial and governmental domains. These systems enable many applications, such as identifying individuals on social media, locating missing persons, assisting in law enforcement and surveillance activities, and authenticating personal identities [207, 208]. This rapid adoption benefited from significant advances in face recognition systems, such as Amazon Rekognition, and the wide availability of labeled facial datasets [209, 26].

While these systems are often evaluated for the average accuracy on available datasets, more is needed to learn about their robustness and fairness against distributional shifts in input data. As such systems are deployed in applications with significant societal impact, ensuring their ethical and reliable use is essential. We posit that characterizing and improving the performance of the deployed

systems requires understanding how they make decisions. Towards that end, counterfactual explanation techniques aim to generate human-understandable input modifications that would have resulted in a different output decision by the face recognition system. By auditing which attributes of the input data were most influential in the system's decision-making process, counterfactual explanation techniques can identify unintended biases and failure modes in face recognition systems. These explanations would also provide insights into improving face recognition performance, which enables a more transparent deployment.

In this chapter, we propose a new method to generate counterfactual examples for face recognition systems. The largest hurdle towards counterfactual analysis of existing face recognition systems is generating realistically modified faces that cover a range of demographic and semantic attributes. Sampling natural inputs that satisfy this condition, such as faces with different skin tones, lighting conditions, hairstyles, or accessories, is nearly impossible. We address this limitation by utilizing recent innovations in Text-to-Image generative models to semantically change faces until the face recognition system reaches a different decision. This method involves generating a large set of identities with diverse demographic attributes, generating multiple face images for each identity, and applying different semantic attributes to the generated faces.

Text-to-image (TTI) diffusion models have become popular due to their unprecedented image-generation capability. Taking a textual prompt as input, these models generate realistic images that align with user intentions. Their ability to synthesize and modify human faces has spurred research into using generated face images in training data augmentation and model performance assessments [210, 211]. For example, face recognition systems can benefit from synthetic datasets that exhibit more demographic diversity than existing natural datasets [212, 213].

This chapter analyzes the quality of synthetic datasets for facial recognition applications and whether they exhibit demographic disparities. Achieving this objective requires generating a large set of identities belonging to diverse demographic groups and generating multiple (different) faces for each identity. Existing diffusion models are incapable of meeting this objective for two reasons. First, aligning the generated faces with the provided prompt is challenging [214, 215]. Second, generating multiple faces with the same identity in a one-shot fashion is typically infeasible [214, 215]. Limited research exists in this space. Previous works either optimize the diffusion model to a particular demographic group,

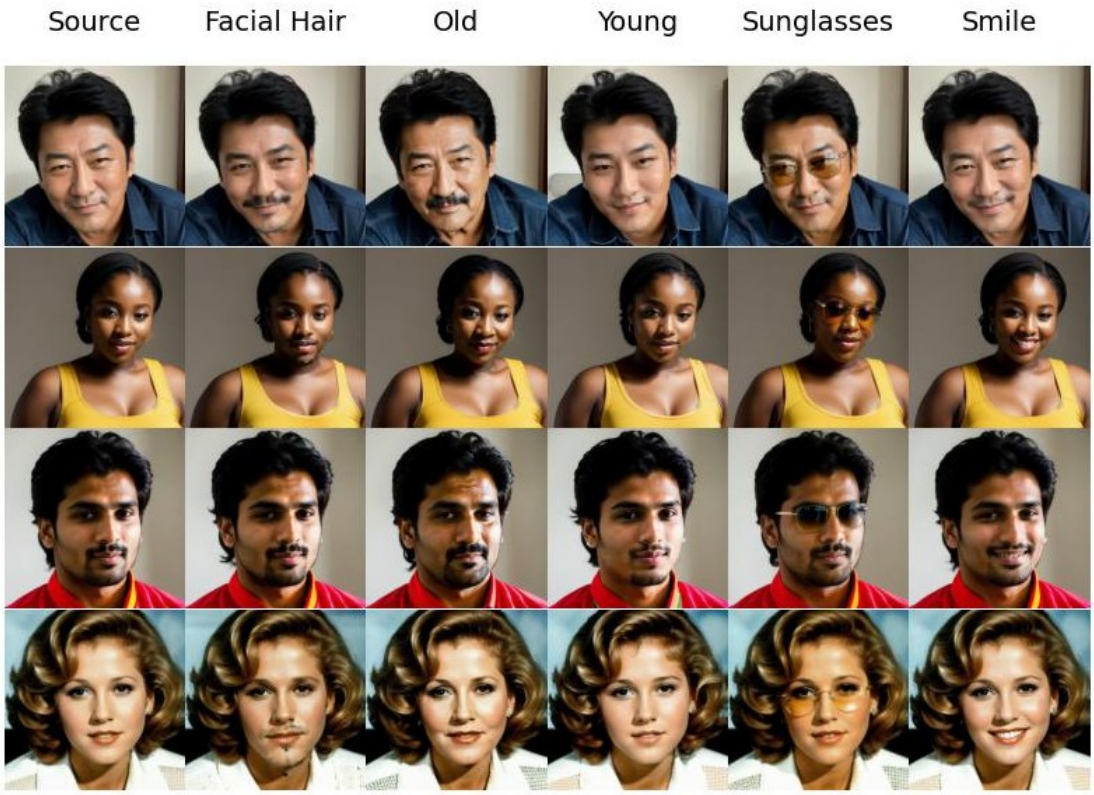


Figure 5.1: Samples of the non-celebrities dataset using Realism model for four demographic groups: 'East Asian Male', 'Black Female', 'Indian Male', and 'White Female'. 'Source' refers to images generated from Realism using the prompt template. The second to sixth columns show transformed images when an attribute is applied to 'Source' using SEGA.

generate faces without a notion of identity, or limit their objectives to frequency analysis of the demographics of the generated images [216].

In this chapter, we propose a new framework to generate synthetic face images, as shown in Fig. 5.1. Our face-generation pipeline takes as input demographic attributes, applies custom prompts to generate identities for each demographic attribute, and utilizes image editing models [215] to generate diversified faces for each identity. The resulting dataset, which we manually verify, resembles a natural face image dataset, albeit demographically balanced by design.

We then apply a three-pronged approach to assess the synthetic face image dataset's quality through face verification [217], quantitative quality metrics [218], and a user study. Our evaluation shows that generated images exhibit demographic disparities in the eyes of face recognition systems. Results from our user studies

show a disparity in the quality of the generated faces for different demographics, with images belonging to majority demographics rated as higher quality. We also study the efficacy of edit correctness metrics built on CLIP and DINO [219, 220]. We find these metrics do not correlate with human preferences in facial semantic changes. Research is needed to develop perceptually aligned metrics. Finally, our findings suggest that methods intended to mitigate bias exhibit demographic disparities in the quality of generated images.

To the best of our knowledge, this chapter is the first work that: (1) *provides an end-to-end pipeline*, utilizing a TTI diffusion model, to generate batches of synthetic faces annotated with fine-grained attributes; (2) *evaluates the quality of large-scale batch-generated faces* using a user study; and (3) *assesses the fidelity* of recently proposed TTI quality metrics on face images.

5.2 Related Work

In the following, we describe recent works in the context of synthetic face image generation and associated biases.

Synthetic Face Image Generation

TTI diffusion models, such as DALL-E [221] and Stable Diffusion [222], rely on internal randomness to generate high-quality examples through denoising steps. They employ CLIP [223] or its variants as text encoders. Thus, a text prompt is sufficient to control the output of a TTI diffusion model.

Two challenges arise in prompt-based face generation. The first is aligning the generated faces with the provided prompt [214, 215]. The second is generating multiple faces belonging to the same identity in a one-shot fashion [214, 215]. There exist methods to better control image generation. These methods include segmentation masks and inpainting [220]; text-inversion, which learns a text token that corresponds to certain image concept [224]; model fine-tuning and embedding optimization [225]. While these techniques are generally effective, they are unsuitable for large-scale generation of diverse faces. They often disrupt the fast and natural interface that differentiates TTI diffusion models.

In this chapter, we aim to analyze the synthetic faces generated by TTI diffusion models. This objective requires generating a large set of identities belonging to

diverse demographic groups and generating multiple (different) faces for each identity. We devise a novel pipeline that employs semantic guided attention (SEGA) [215], fixed seeds, and specialized prompts. The pipeline, described within the Framework section, depends on neither inversion nor fine-tuning.

Bias in Face Image Generation

Recent works [213, 226, 212] have studied the bias of TTI face generation by analyzing the proportions of demographics in generated images. Seshadri et al. found that generative models amplify the discrepancies in training data [226]. One example is gender-occupation bias, where Stable Diffusion can generate highly biased face distributions from a gender-neutral prompt about occupations. Friedrich et al. mitigated these biases with a post-processing technique called Fair Diffusion [213]. When the user inputs their prompt, a model detects the potential bias in the prompt and steers the output to a fairer region, leveraging a lookup table of instructions and the semantic image-editing technique SEGA [215]. Similarly, Smith et al. utilized InstructPix2Pix [219], an instruction-based image editing model, to edit existing images to be demographically balanced. While this dataset debiasing technique results in finer-grained control over demographic attributes, it introduces a distribution shift between natural and synthetic images. It also stacks the biases of different models [212].

Luccioni et al. performed a different bias characterization that relies on correlating model outputs in the embedding space with social attributes [227]. The authors found three popular TTI models are biased toward masculine and white concepts. Struppek et al. studied another source of bias resulting from non-Latin scripts [228]. They found that using special non-Latin characters better exposes the internal biases of models and proposed using homoglyphs to mitigate this bias. Muñoz et al. analyzed the bias in relatively older face generation models trained on the CelebA and FFHQ datasets [229]. Using quantitative metrics, including demographic frequencies, face recognition verification, and Fréchet inception distance, they found that the generative models are biased.

These conclusions are consistent with earlier GAN literature, where Maluleke et al. found them to generate racially biased distributions of faces [230]. Maluleke et al. went one step further by analyzing the quality of generated faces through a user study, where generated faces from minority groups (e.g., Black) exhibited

lower quality.

In summary, existing works focus primarily on frequency analysis to characterize bias in TTI models, propose embedding-based metrics to evaluate the quality of generated images, and utilize synthetic data to mitigate bias. In our work, we characterize the synthetic datasets, showing that methods intended to mitigate bias exhibit demographic disparities in the quality of generated images. We go beyond frequency analysis by rating image quality in a user study. We also utilize the user study results to assess embedding-based metrics in characterizing the quality of the generated images.

5.3 Framework

We develop a framework, as depicted in Fig. 5.2, to audit the characteristics of generated face images. This framework consists of choosing the demographic conditions, prompting diffusion models to generate identities according to these conditions, followed by evaluating the generated images both quantitatively and qualitatively.

Notation

We first prescribe the notation used within this chapter. There exists a sample space $\mathcal{X} \subseteq \mathbb{R}^d$ and label set $\mathcal{Y}$. A sample $\mathbf{x} \in \mathcal{X}$ is a d-dimensional vector. If $\mathbf{x}$ is an RGB image, then d equals $3 \times h \times w$, which corresponds to the number of channels multiplied by the number of pixels in the image. In the context of face recognition, we assume each face image $\mathbf{x}$ depicts an identity $y \in \mathcal{Y}$. A face recognition model $f : \mathcal{X} \to \mathcal{Y}$ is trained on a finite dataset $S \in \mathcal{X} \times \mathcal{Y}$. S is drawn i.i.d. from distribution $\mathcal{D}$. Sometimes, when clear from context, S refers to an unlabeled dataset. A metric embedding network $f_k : \mathcal{X} \to \mathbb{R}^k$ is often internal to deep-network based classifiers. Metric embedding network f_k maps inputs to a k-dimensional embedding space. If two samples have low pairwise distance, they are assumed to be more similar in the associated label space.

We analyze disparities in generative models across social attributes. To analyze these disparities, we examine synthetic face image quality and the performance of generated images in face recognition tasks. A common class of social attributes is demographics. With respect to demographics, we use terminology consistent

with Buolamwini and Gebru [17], work among the most cited in the space of face recognition fairness. Face images are annotated by sex and ethnicity. Sex annotations are "Male" and "Female." Ethnicity annotations are "White," "Black," "East Asian," and "Indian." The set of demographic groups is denoted as G, where g is a placeholder for a demographic group in G. In this chapter, we study eight demographic groups, corresponding to sex-ethnicity combinations.

Given a text prompt p from the space of prompts $\mathcal{P}$, a text-to-image model $h_q : \mathcal{P} \to \mathcal{X}$ returns the image prescribed by its textual prompt p, where the random seed q is a real number. Because diffusion models have internal randomness, each q generates a different realization of the same prompt p. In our framework, we encode the identity y and its demographic group g in the textual prompt p, and we vary the seed q to generate multiple images of the same identity. We use a fixed set of seeds to ensure the reproducibility of generated images.

Generative Models

We generate synthetic faces using two TTI Diffusion models: the open-source Stable Diffusion v2.1 model by Stability AI and the finetuned Realistic Vision Model[1], hereafter referred to as *SDv2.1* and *Realism*, respectively. We analyze the images generated by these models individually to assess their efficacy in face-generation pipelines.

SDv2.1 is finetuned from the Stable Diffusion v2 (SDv2) checkpoint, which was trained from scratch on a subset of the LAION-5B dataset. SDv2.1's training dataset contains more faces than that of SDv2[2]. Hence, SDv2.1 performs better in generating faces than SDv2. Realism is among the many openly available fine-tuned models from the checkpoints of Stable Diffusion. However, its exact implementation details are not known. We treat both SDv2.1 and Realism as grey-box models. Both models are capable image generators with differing performance characteristics, and our framework is agnostic to their implementation details. The design of the system shown in Fig. 5.2 can be used with any relevant text-to-image generative model to synthesize scalable batches of facial data useful for training data augmentation or as tailored test sets for face recognition applications.

[1] `https://huggingface.co/SG161222/Realistic_Vision_V4.0_noVAE`

[2] `https://stability.ai/blog/stablediffusion2-1-release7-dec-2022`

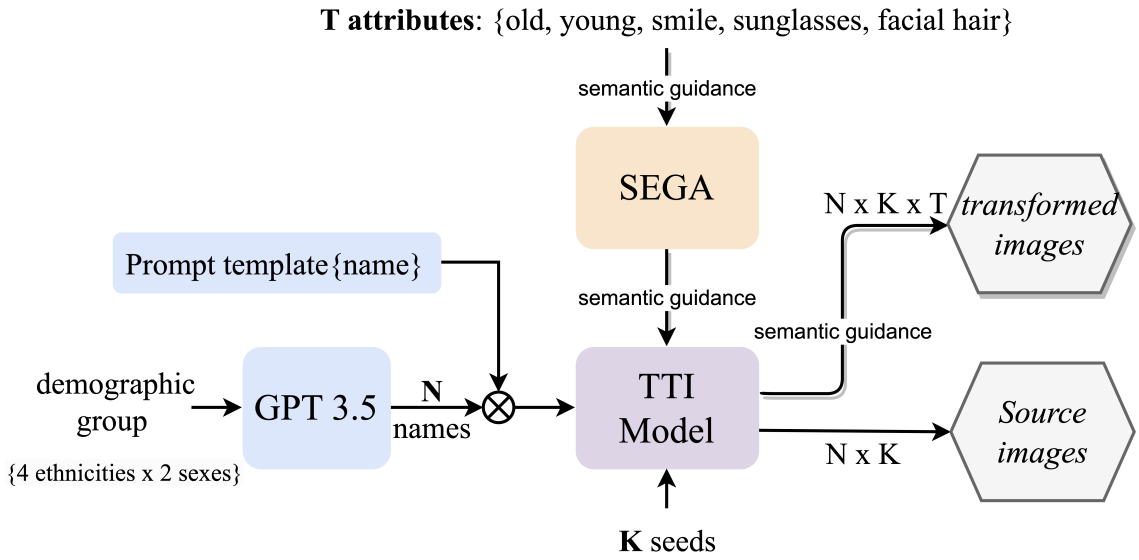


Figure 5.2: Our data generation pipeline: (1) generate N names (identities) belonging to each demographic group g ∈ G and insert them into the prompt template p{name}, (2) TTI generates K images per identity, using K seeds, (3) SEGA steers the TTI generation to incorporate each of the T semantic attributes.

To diversify generated faces, we employ the semantic-guidance image generation technique SEGA [215]. SEGA steers the TTI model towards generating images that incorporate semantic concepts based on user-provided textual edits while keeping the rest of the image semantics intact, all without the need for fine-tuning the TTI models. This technique proves valuable in creating faces with diverse attributes, such as incorporating sunglasses. Moreover, recent works [213, 212] leverage SEGA and similar methods for fair face image generation by introducing demographics as semantic concepts during image generation. Thus, we study the efficacy of incorporating SEGA in the image generation pipeline.

Data Generation Pipeline

To generate our facial datasets, we design a prompt that specifies a demographic group and an identity associated with that group. The prompt guides the model to generate a set of diverse face images for each of these identities.

Identity. We found that when we explicitly mention the demographic group in the prompt, like *an Indian man*, the generated images exhibit limited diversity; i.e.

identities look quite similar. To encourage the generation of more varied identities, we employed *names* as indicators of different identities within demographic groups. We found that TTI models interpret names as proxies for ethnicity and sex, and each name carries a unique identity despite the randomness of the generation process.

For each of the eight demographic groups we study in this chapter, we instructed GPT-3.5 to create two separate lists of names—one comprising 'celebrity' names and the other 'non-celebrity' names. For the non-celebrity (celebrity) collection, we generated 20 (30) names per demographic group. The two lists reflect different levels of knowledge within the TTI model: celebrity images are more likely to exist in the training data of TTI, while non-celebrities are more likely to be indirectly learned by the model.

Prompt. We desire prompts that guide the model to generate multiple and diverse face images with user-specified semantics. Including a name within a prompt encodes both identity and demographic information. Trial and error, combined with our user study, led us to the below approach.

For Realism, we experimented with a set of prompts, and we found this template to generate face images of high quality: "A photo of the face of {identity}." We vary the TTI generator seed to generate multiple images per identity and prompt. We also add a set of *negative* prompts that steer the model away from unrealistic, cartoon, or low-quality image generation. These negative prompts are frequently used in face image generation. For a fair comparison, we use this prompt template along with a set of pre-selected five seeds to generate images of all identities and demographic groups.

For SDv2.1, we observed that the prior template generates images of poor quality on both celebrity and non-celebrity identities. Thus, we expanded the prompt template as follows: "A photo of the face of ({identity}:2.0). (realistic:2.0). (Face shot only:2.0)." This revised prompt improved the generated image quality of celebrity identities. However, it did not have the same effect on non-celebrity images. Thus, we decided to evaluate only the celebrity identities for the SDv2.1 model.

We manually validated that the generated images from both models contain a face image, different seeds generate diverse images of the same identity, and that identities are distinct and belong to the intended demographic group.

Attributes. Using SEGA with Realism and SDv2.1, we induce five attributes to the generated data: 'young', 'old', 'facial hair', 'sunglasses', and 'smile.' We refer to the images obtained without SEGA as *source images* and with SEGA as *transformed images*. All the synthesized images are of 512×512 resolution. For SDv2.1, to ensure better quality, we generate the images at 768×768 and then downsample them to 512×512. Fig. 5.1 shows a sample of the non-celebrity images synthesized using Realism and SEGA.

In total, we generate 800 source and 4000 transformed non-celebrity images, and we generate 1200 source and 6000 transformed celebrity images per model.

Evaluation Methods

We use three independent evaluation methods to assess the quality of the generated datasets: quantitative metrics, face verification, and user study.

Quantitative Metrics

The metrics below are used to evaluate the overall quality of the source and transformed images.

- **Image-Image Metrics:** These are mainly used to verify identity retention under SEGA transformation. CLIP-I and DINO-I measure the cosine similarity between the source and transformed images' CLIP [223] and DINO-v2 [231] embeddings, respectively. Higher similarity implies that the identity is preserved.

- **CLIP-directional**: CLIP-directional [232] intends to identify the correctness of the semantic change in the transformed image. It measures the similarity of the change between the embeddings of the source and transformed images and the change between their captions.

Face Verification

Face verification accuracy utilizes pairwise face comparisons to measure embedding space quality. The embeddings of two faces depicting the same identity are expected to be close to each other. We analyze face verification performance on Facenet [217], a well-studied face recognition network.

Our analysis of face recognition models focuses on verification accuracy. Given two face images x, x', verification accuracy VER is computed as:

$$\begin{aligned} \mathrm{VER}(x, y, x', y') \triangleq \mathbb{1}[y = y'] \cdot \mathbb{1}\left[\rho(f_k(x), f_k(x')) < t\right] \\ + \mathbb{1}[y \neq y'] \cdot \mathbb{1}\left[\rho(f_k(x), f_k(x')) \geqslant t\right] \end{aligned} \tag{5.1}$$

where $\mathbb{1}$ denotes the indicator function and threshold t is chosen heuristically to minimize false verification rate. Further, y and y' are identities associated with x and x', respectively. We report the average verification accuracy as computed across sets of pairs. Within our evaluation, sets of pairs are constructed so that half of the pairs correspond to the same identity. When analyzing verification accuracy for the user study, we implicitly assume that humans can perfectly distinguish the identities of generated faces.

To study the effect of demographics on verification, we report two notions of verification accuracy: same group and any group. For a specified group g, same group verification accuracy refers to the evaluation of VER on lists of pairs in which both images x, x' belong to group g. Any group verification accuracy refers to the evaluation of VER where only each pair's first image x must be in group g.

We utilize the Labeled Faces in the Wild (LFW) dataset as a baseline for natural face verification. LFW is a canonical dataset for face recognition tasks. The LFW dataset contains 13233 images and a total of 5749 unique identities. Demographic annotations for images in LFW were obtained from the system introduced by Kumar et al. [23].

User Study

We conducted a human evaluation of the generated images from both models combined with SEGA. Toward that end, we designed an online Qualtrics survey for each model-identity collection pair, resulting in three surveys: (SDv2.1 Celebrities, Realism Celebrities, and Realism Non-Celebrities). The surveys are approved by our IRB and are conducted on the Prolific platform.

For each survey, we randomly sampled 15 identities per demographic group, one image per identity; 120 images in total. We paired each source image with its 5 transformed images corresponding to the 5 semantic attributes. This results in 600 source-transformed image pairs per survey. We presented each participant with a set of 21 blocks, along with at least one attention question. Each block shows two

images: one source image (without SEGA), and one transformed image (using SEGA), along with the transform instruction used by SEGA. For each block, the participant answers three questions: (1) whether the two images depict the same person, (2) the consistency of the transformed image with the edit instruction on a 5-point scale, and (3) how they rate the quality of the two images on a 5-point Likert scale. An example block is shown in Fig. 5.3b.

Fig. 5.3 presents a snippet of the survey design. The survey instructions (Fig. 5.3a) encourage participants to focus solely on the correctness of the edits and to disregard any violations of social norms they may observe in the images. This instruction is important since our work's ultimate goal is to generate a dataset that facilitates an assessment of face recognition applications in both in-distribution and out-of-distribution scenarios. Out-of-distribution images may possess attributes that seem unusable to participants.

For each survey, we recruited 85 participants, and each image pair received three ratings on average. Each participant was compensated $3.5 for their effort, with an average completion time of 15 minutes. The study was distributed evenly to male and female participants.

5.4 Evaluation

After generating the datasets, we apply the evaluation methods to analyze the associated demographic discrepancies. Three questions guide this evaluation:

Q1. How does face verification on synthetic data compare to natural data and does it exhibit demographic disparities?
Q2. Does the quality of synthetic face images depend on the demographic group?
Q3. Can quantitative metrics replace expensive user studies to assess the quality of synthetic face images?

Face Verification

Face verification performance is depicted in Fig. 5.4. The figure shows the verification accuracy measured on LFW and synthetic datasets. Across all demographics and datasets, with one exception, we observe that generated faces perform worse than natural faces (LFW). Only in the White demographic does a synthetic dataset,

(a) Survey Instructions

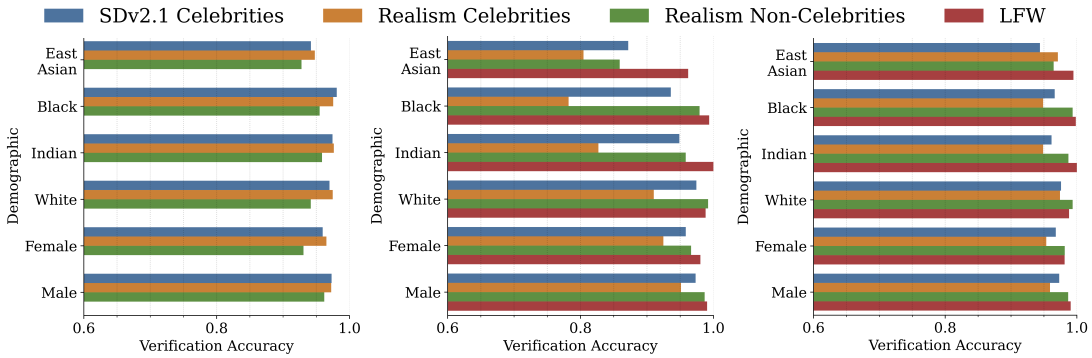
(b) An example of a survey block

Figure 5.3: User survey instructions and example block of questions.

Realism Celebrities, have better face verification performance than natural data. We also observe that for each demographic and dataset pair, same-demographic verification accuracy is often notably less than its any-demographic counterpart. Hence, we conclude that face recognition systems are demographically aware on generated faces in a manner similar to natural faces.

Synthetic Face Image Quality

Table 5.1 presents the average survey scores in terms of image quality and transformation correctness across all demographics and datasets. The scores suggest that



(a) Survey Responses (b) Same Demographic Pairs (c) Any Demographic Pairs

Figure 5.4: Verification Accuracy is plotted across four datasets. Each row is a demographic, and each dataset is depicted with a different hue. Note that each plot is x-axis limited between 0.6 and 1.

image quality depends on the identity's demographic group. Moreover, SEGA transformations drop the quality of all images, and the drop is also demographic-dependent. We use one-way ANOVA in an attempt to reject null hypotheses of forms:

Null Hypothesis 1. *The per-demographic distributions of source image quality in ⟨Dataset⟩ are identical.*

Null Hypothesis 2. *The per-demographic distributions of transformed image quality in ⟨Dataset⟩ are identical.*

Null Hypothesis 3. *The per-demographic distributions of quality difference between source and transformed images in ⟨Dataset⟩ are identical.*

On the Realism Non-Celebrities and Realism Celebrities datasets, one-way ANOVA rejects null hypotheses 1 to 3 with p-values less than 0.05; corresponding p-values appear in Table 5.2. This test tells us that for these two datasets, source image quality, transformed image quality, and the difference between source and transformed image quality have a dependence on demographics. The only dataset for which image quality does not conclusively depend on demographics is SDv2.1 Celebrities.

The same observation of demographic dependence applies to the transformation correctness measures (M3, M4). It is interesting to note that demographic groups that have higher source image quality are not consistent with groups of

	Dataset	Demographic group					
		E Asian	Black	Indian	White	Female	Male
M1	D1	4.463	4.401	**4.394**	**4.515**	4.428	4.459
	D2	**4.253**	4.240	**4.045**	4.097	4.166	4.140
	D3	4.121	**4.149**	4.112	**4.039**	4.112	4.099
M2	D1	0.195	0.122	0.187	0.184	**0.244**	**0.098**
	D2	0.190	**0.085**	0.114	**0.592**	0.364	0.140
	D3	0.100	**0.156**	0.123	0.123	0.154	**0.098**
M3	D1	4.020	3.972	**4.144**	**3.945**	3.954	4.092
	D2	3.624	3.367	**3.717**	**2.636**	3.203	3.455
	D3	**3.747**	**3.197**	3.516	3.325	3.492	3.412
M4	D1	87.5	84.6	**90.3**	**83.8**	85.5	87.7
	D2	78.4	69.5	**80.7**	**51.2**	66.0	73.6
	D3	**79.9**	**65.2**	72.3	69.9	72.2	71.8

Table 5.1: User survey average answers to the following measures: M1: source image quality on a 5-point scale, M2: drop in image quality after SEGA transformation, M3: SEGA transformation correctness on a 5-point scale, and M4: percentage (%) of correct transformation (transformation correctness score $\geqslant 3$ out of 5). D1: Realism Non-Celebrities, D2: Realism Celebrities, D3: SDv2.1 Celebrities. Highest and lowest scores are highlighted in **bold**. E Asian denotes the East Asian demographic group.

Dataset	Null hypothesis 1	Null hypothesis 2	Null hypothesis 3
SDv2.1 Celebrities	0.498	0.573	0.488
Realism Celebrities	0.000474	6.31×10^{-25}	4.02×10^{-28}
Realism Non-Celebrities	0.0306	5.16×10^{-5}	2.47×10^{-5}

Table 5.2: p-values associated with one-way ANOVAs on null hypotheses 1 to 3 for Realism Celebrities, Realism non-Celebrities, and SDv2.1 Celebrities datasets

higher transformation correctness. This suggests that SEGA introduces its own biases in the generative pipeline.

Quantitative Metrics vs. User Study

User studies are the most direct way to measure human perception of generated faces. Unfortunately, they are prohibitively expensive when implemented at scale. If we have a metric serving as a proxy for human sentiment toward generated face quality, costs associated with generating realistic face data could be drastically

Dataset	Null hypothesis 4		Null hypothesis 5
	CLIP-I	DINO-I	CLIP Directional
SDv2.1 Celebrities	0.147	0.107	0.128
Realism Celebrities	0.197	0.122	0.348
Realism Non-Celebrities	0.245	0.142	0.0908

Table 5.3: Spearman correlation coefficients for null hypotheses 4 and 5. Each correlation coefficient is statistically significant.

reduced. We analyze the correlation between the different metrics and the questions posed in the user study regarding the quality of the source and transformed images, the presence of semantic change, and identity retention after applying the semantic change. We calculate the Spearman correlation coefficients between the metrics and the scores to the user-study questions and once again make use of one-way ANOVA tests to reject null hypotheses:

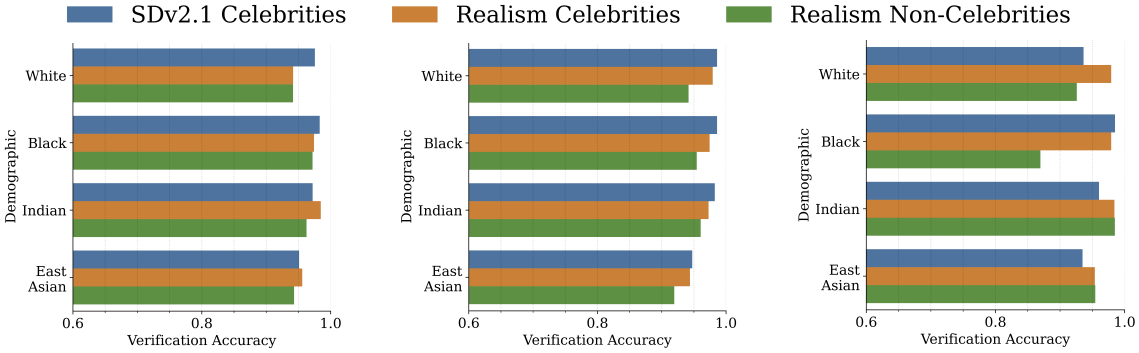
Null Hypothesis 4. *On ⟨Dataset⟩, there is no monotonic relationship between image-image ⟨similarity metric⟩ and maintenance of identity post application of semantic change.*

Null Hypothesis 5. *On ⟨Dataset⟩, there is no monotonic relationship between CLIP-directional and appearance of the semantic change.*

On all three datasets, one-way ANOVA tests enable us to reject null hypothesis 4 on image similarity metrics CLIP-I and DINO-I. We also similarly reject null hypothesis 5 on the CLIP Directional metric. Despite rejecting null hypotheses, each Spearman coefficient is low, as evident from Table 5.3 and the corresponding p-values in Table 5.4. Hence, in the context of face recognition, image quality metrics are not a suitable proxy for humans in performing both identity verification and transformation verification tasks. This result is partially surprising: DINO-I metric, unlike CLIP-I metric, is designed to recognize differences between images of similar descriptions [218]. Yet, our findings indicate a low correlation between this metric and human assessment.

Dataset	Null hypothesis 4		Null hypothesis 5
	CLIP-I	DINO-I	CLIP Directional
SDv2.1 Celebrities	6.264×10^{-18}	7.35×10^{-14}	3.55×10^{-10}
Realism Celebrities	5.74×10^{-32}	1.022×10^{-100}	3.27×10^{-13}
Realism Non-Celebrities	3.26×10^{-60}	2.18×10^{-9}	6.64×10^{-21}

Table 5.4: p-values associated with null hypotheses 4 and 5.



(a) Black (b) White (c) Hispanic

Figure 5.5: User Verification Accuracy. The y-axis captures queried image demographics. Each subfigure depicts a respondent demographic. Note that each plot is x-axis limited between 0.6 and 1.

5.5 Discussion

Our user study provides direction for follow-on research relating to the Own Race Effect (ORE). ORE refers to the documented tendency of individuals to better recognize faces from within their racial group [233, 234]. We observed that our user survey seems to disagree with ORE as shown in Fig. 5.5. Hence, a rigorous study of perceived identity of images under semantic transformations would be of research value.

As evidenced by the user study, mechanisms of human face perception present unique challenges to the application of generative models in face recognition. Moreover, automated prompt design strategies require access to a metric quantifying the quality of generated images. This does not detract from techniques evaluating generative model performance, rather, it opens a new research avenue: tuning face quality metrics to better align with human preferences. Our techniques apply outside of the face recognition task. Generative models for both audio and text are

equally ripe for performance evaluation.

Though our analysis techniques generalize to other domains, they assume CLIP functions as intended. Unfortunately, CLIP and similar semantic-visual embedding models are trained on internet data. Hence, their embedding space contains biases. Further, CLIP is known to have trouble constructing embeddings for uncommon or otherwise niche words and phrases. Niche words and phrases, such as "inter-eye distance" and "eyebrow slant", which can affect human perceived face identity [235], are problematic for CLIP. Further analysis of semantic-visual embeddings is necessary to gain a full picture of text-to-image generative models. Additionally, CLIP's understanding of cultural constructs is not entirely understood. For example, it is unclear what an "intelligent face" or "beautiful face" means to CLIP. Thus, the semantic transformations we study are explicit face attributes.

Finally, our study does not negate the value of generative approaches in model analysis. Generative examples can serve as a targeted curated dataset. Tailored generation has the potential to mitigate inherent biases found in existing datasets; however, the effectiveness of this approach is closely tied to the data used to train the generative model. It's important to remember that the generated examples are not i.i.d. samples from the natural distribution. Instead, they represent i.i.d. samples from a possibly skewed estimate derived from a finite pool of realized examples within the training set.

5.6 Conclusion

Generative models have been the subject of much recent societal interest. Synthesized examples achieve near-realistic quality. Though recent advances have increased the expressive power of generative models, their performance characteristics remain opaque, especially for face image generation. We put forth a new framework to synthesize diverse face images and evaluate them from multiple perspectives. This chapter demonstrates the need for further research into the properties of semantic-visual embeddings and human perception mechanisms upon generated faces. It further emphasizes the challenges of altering visual data while preserving the semantics of the image. Looking ahead, we aim to investigate comparable approaches for audio applications.

Chapter 6

Conclusion

In this thesis, we examine the privacy, security, and fairness challenges inherent in voice-based and visual-based machine learning (ML) applications, which engage billions of users daily. Specifically, our focus is on three key voice-based applications: speech recognition, keyword spotting, and speaker identification. In the realm of visual-based applications, we concentrate on face recognition, acknowledging its critical role and widespread deployment. For each of these applications, we identify the most significant threat models, quantify the associated risks, and propose solutions aimed at enhancing the trustworthiness of the technology. We outline the contributions of this thesis in the subsequent section and suggest potential directions for future research, building upon the contributions laid in this work.

6.1 Future Research

This chapter describes future research directions that are motivated by this thesis.

Speech Technologies Performance Disparities

Motivation. Major Cloud providers offer speech processing APIs that are highly accurate and convenient to use as a service. These APIs are currently deployed in various systems and applications that are used by millions of users around the world. These APIs perform very well on *the average test case*. However, this performance does not hold for all demographic subgroups. A recent work [236] shows that speech-to-text APIs exhibit a racially biased performance against black

speakers. For example, Apple, IBM, Google, Amazon, and Microsoft APIs word error rates (WER) are substantially higher for black speakers than white speakers. On average, black speakers WER is 35%, while it is 19% for white speakers. Another disparity is observed across the gender attribute, where a higher performance is observed for female speakers. Other works [237, 238, 239, 240] have confirmed this disparity between different population subgroups across the gender, age, skin tone, accents, and geographic locations attributes. Although conventional training datasets are unbalanced with respect to the different population subgroups, Liu et al. [237] found that fine-tuning speech recognition models on a balanced dataset does not significantly reduce the model's WER gap. This experiment shows that speech technologies need a thorough investigation to understand the underlying causes of performance disparities. Moreover, similar disparities have been observed in vision [17] and language models [241, 242, 243]. These disparity commonalities across different tasks and modalities suggest that the ML community makes similar mistakes along the ML pipeline such as data collection and processing, the design of training algorithms, and the way in which performance metrics are reported.

Privacy Implications. The performance disparities of these APIs not only deteriorate the underrepresented groups' user experience but also increase their vulnerability to privacy violations and security risks. For example, a performance disparity in the KWS model will lead to a higher false activation rate for the underrepresented group. Thus, this disparity puts them at a higher risk of leaking their private conversations and maliciously controlling their VA. Moreover, this performance discrepancy may open a new attack angle for the adversaries. Another example is voice-based speaker authentication where the system has lower accuracy in identifying the underrepresented group users. These users are more susceptible to spoofing attacks and, thus, are at a higher risk of fraud and impersonation.

Analogous to the discussion in Chapter 5, an essential research direction involves examining spurious correlations and failure modes in speech technologies. This investigation represents an initial step towards designing unbiased speech technologies.

Unauthorized Collection of Voice Biometrics

Motivation. Speaker identification serves as an intuitive and valuable authentication solution for a variety of applications, including phone banking, customer verification in call centers, voice-based check-in systems, and personalization of multi-user smart devices. This value is derived from the technology's convenience for customers, its reliability, and the cost reduction it offers businesses. Speaker identification employs the speaker's voice biometrics as proof of identity. Initially, the application enrolls customers into the speaker identification system. During this phase, the API extracts the speaker's voiceprint and stores it in the cloud. Subsequently, at runtime (inference), the API compares the voice features of the speaker against the stored voiceprint to identify and authorize the individual. However, current services are susceptible to various failure modes, such as impersonation attacks, and other malicious uses. A particularly concerning, yet under-researched, malicious use case is speaker surveillance. This vulnerability arises from the service's failure to validate user consent for enrollment and identification.

The abundance of speech data available on social media platforms lowers the par for malicious individual surveillance; it allows adversaries to enroll any arbitrary speaker into a speaker identification service without their permission or consent. Utilizing this powerful API, an adversary can then identify speakers across all recorded media on the internet, effectively tracking their speech and opinions. This threat becomes increasingly pronounced with the rising interest in smart cities, where microphones are deployed in both public and private spaces and speech is used to interact with augmented reality environments. Potential adversaries include surveillance agencies that could track individuals' private and public conversations about political and religious views for purposes of retaliation or social ranking. Another possible adversary is an advertising agency using this capability to monitor conversations for targeted advertising. This threat is facilitated by the fact that the API does not require validation of the speaker's consent for enrollment and identification, leaving speakers with no control over their speech and voice biometrics.

Moreover, the advancement in speech synthesis technology has reached a point where fake speech is nearly indistinguishable from natural speech. This development opens up the possibility for adversaries to impersonate other speakers using

synthetic voices, thereby gaining unauthorized access to security-critical systems. For instance, an adversary could deceive a phone-banking system that utilizes a speaker recognition API for authentication, resulting in unauthorized access to a user's bank account. This scenario highlights a significant vulnerability where the system, misled by the synthetic voice, erroneously authorizes the adversary.

Proposed Future Work. In future work, we propose a dual-pronged solution to mitigate the aforementioned threats, focusing on ensuring that enrollment and identification requests originate from a *live* speaker who provides explicit *consent* for identification. Firstly, we introduce Tainted-Speech, a user-side system designed to protect users' uploaded speech on social media and video platforms. Tainted-Speech applies adversarial perturbation to the user's speech, effectively 'tainting' the speaker's voice from the perspective of the speaker recognition API, thereby preventing the extraction of a voiceprint. This approach serves as a safeguard against unauthorized speaker enrollment and identification.

Secondly, we propose enhancing the speaker recognition system with an audio-captcha system. This audio-captcha functions as a challenge-response audio mechanism, verifying two crucial aspects: that the speaker is a live human rather than a synthetic voice, and that they are intentionally submitting their voice to the API. By integrating this additional layer of security, the system can more effectively discern and validate genuine users, bolstering its defenses against unauthorized access and exploitation.

Utilize Synthetic Face Images for CounterFactual Explanation

Auditing ML is Challenging. The inner workings of neural networks are neither human-understandable nor theoretically tractable. Consequently, the best way to understand model performance is empirical validation. In traditional machine learning settings, practitioners understand the model performance by evaluating it on a validation set. Often, this validation set follows the same distribution as the training data. This approach provides no insight into performance on out-of-distribution data, which is the common case in real-world deployment.

Counterfactual Explanation. One way to achieve this goal is through explainability methods, which explore the failure modes and spurious correlations inherent

in the model. Counterfactual explanation (CE) is a post-hoc technique that aims to perform hypothetical input modifications that would have resulted in a different decision by the model. By revealing which features of the input data were most influential in the model's outcome, CE identifies unintended biases and spurious correlations and provides insights into how to improve the model's performance.

Recent works [244, 245] have explored diffusion models for generating counterfactual examples, aiming to identify spurious correlation and failure modes in vision classifiers trained on ImageNet. Vendrow et al. [245] defined 23 counterfactual shifts such as 'at night,' 'blue,' 'in the beach,' 'in the snow,' and 'sketch.' They performed textual inversion, a few-shot fine-tuning step, to encode these shifts in the diffusion model. However, this method requires fine-tuning (training) for each new counterfactual concept, limiting its scalability. On the other hand, Wiles et al.[244] explored the failure modes using an automated approach. They utilized a diffusion model to generate a large collection of test images for each label in the classifier under test. Then, they semantically clustered the images that received an incorrect prediction. Using a captioning model, they labeled these clusters as failure modes of the classifier model. However, it's worth noting that neither of the two works explored the application of face recognition models, where demographic disparity and the intricacies of facial features pose unique challenges for counterfactual edits.

Generative AI for Counterfactual Explanations. A follow-up of our work in Chapter 5 is to utilize the generation pipeline to generate counterfactual examples for face recognition models. The methodology is to: (1) create a list of all possible semantic attributes of human faces, (2) utilize SEGA (section 5.3) to incorporate these semantic attributes into the generated faces, and (3) pass the generated images to the face recognition model and analyze the change in model outputs as a function of the change in semantic attributes. By design, this methodology associate each face image with fine-grained ground truth semantic and demographic labels.

Chapter 7

Appendix

7.1 Circular Microphones Spatial Diversity

Here, we evaluate EKOS (Chapter 2) in the setting of a single centralized VA that has $m = 4$ or $m = 7$ microphones — similar to commercial VA devices. We use the circular microphone array model from *Pyroomacoustics* package to generate the RIR at closely located microphone in a circle of 5cm radius, representing the device board.

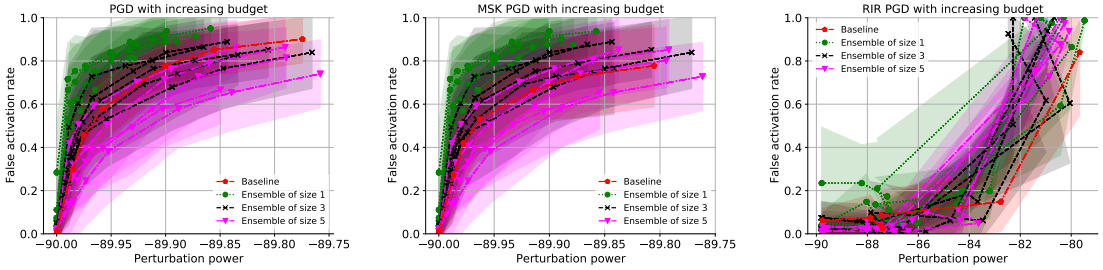
Table 7.1 shows the ensemble natural accuracy of EKOS's pipeline at $l = m = \{4, 7\}$, where l is the number of ensemble models. We assign a model to each microphone, and apply inference time randomness: (1) filter selection and (2) filter cutoffs random shift within $\pm$200Hz. As the table shows, the ensemble accuracy is only slightly lower than the values at Fig.2.3. We observe that the 4 microphones case is slightly better than 7 microphones, probably because they experience more spatial diversity than the 7 microphones setup leading to a higher majority vote accuracy.

7.2 Adversarial Perturbation Imperceptibility

We investigate the relationship between the imperceptibility of the attack and the ensemble size l. In EKOS, a successful attack needs to trick $\frac{l}{2} + 1$ models. Thus, increasing the number of models l requires a higher perturbation size to reach the same attack success rate (false activation rate) as shown in Fig. 7.1.

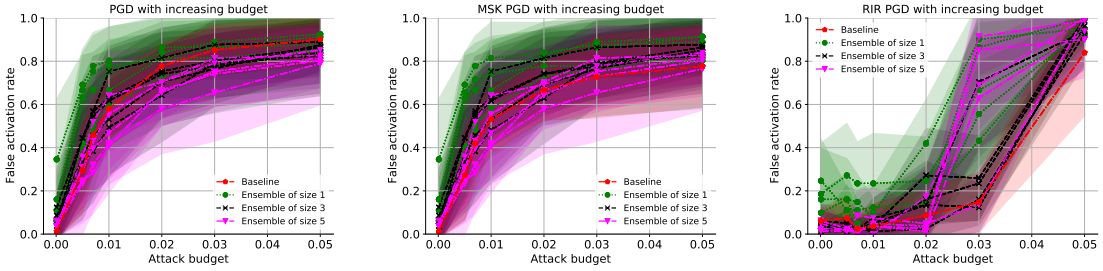
Architecture	Random filters		Random cutoff shift	
	$m = 4$	$m = 7$	$m = 4$	$m = 7$
TC-ResNet8	91.32	87.34	77.02	78.79
TC-ResNet14	92.06	93.04	78.64	77.19
TC-ResNet2D8	90.24	87.51	85.71	82.85
DS-CNN-M	89.18	89.90	74.84	74.56
Random Arch.	89.20	89.10	79.208	84.846

Table 7.1: EKOS's accuracy (%) when deployed on a single VA that has m microphones and $l = m$ models, at different architectures and run-time randomness.



(a) PGD (b) PGD with masking (c) PGD with RIR EoT

Figure 7.1: False activation rate (%) versus the average perturbation power (dB_w) received by EKOS, at ensembles of sizes $l = 1, 3, 5$ along-with a TCResNet8 baseline model under (a) PGD, (b) PGD with frequency mask, and (c) PGD with RIR attacks.

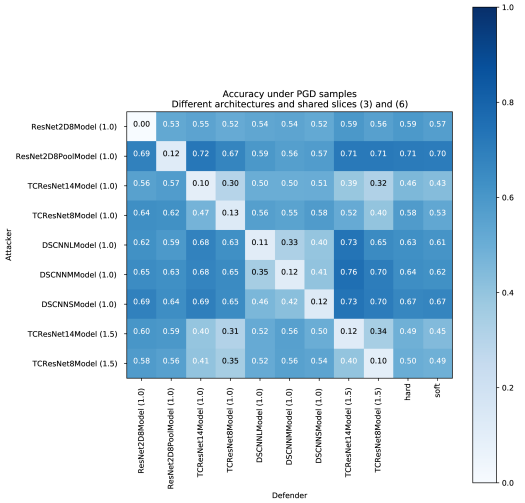


(a) PGD (b) PGD with masking (c) PGD with RIR EoT

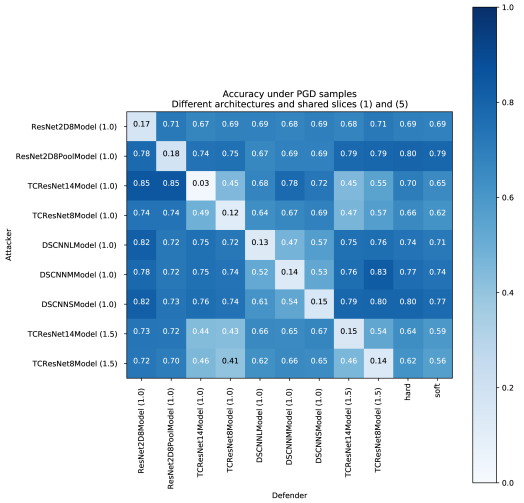
Figure 7.2: False activation rate (%) of EKOS at 5 randomly selected ensembles of sizes $l = 1, 3, 5$ with $\pm$200Hz random cutoff shift, along-with a TCResNet8 baseline model, under adversarial examples generated by (a) PGD, (b) PGD with frequency mask, and (c) PGD with RIR attacks.

Adversarial Examples Transferability

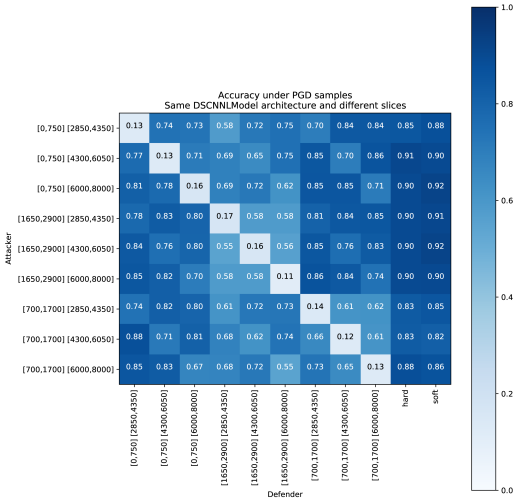
Figure 7.3 shows the transferability map of the PGD attack with 100 epochs and 0.05 perturbation budget across 9 architectures and 9 filters. Note that with this budget, models usually reach near-random guess performance.



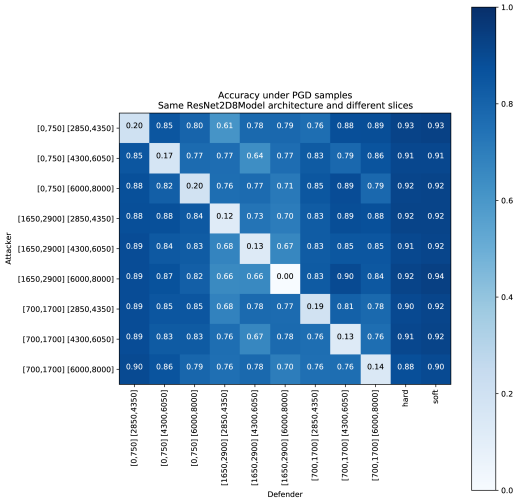
(a) Filter Slice 3 and 6



(b) Filter Slice 1 and 5



(c) DS-CNN-L Model



(d) ResNet2D Model

Figure 7.3: Transferability of PGD with 100 epochs and 0.05 perturbation budget across (a,b) 9 architectures with shared filter slices, and (c,d) 9 filters with a single shared architecture.

7.3 Topic Model Proof

This appendix contains the proof of Theorem 3.4.4 of Sec. 3.4.

Table 7.2: Notations

Symbol	Explanations
$\mathcal{V}$	- the vocabulary
t	- total number of topics
w	- represents a word
k	- minimum number of words per topic
$\mathbb{T}$	- represents a true topic
$\mathbb{T}'$	- represents a perturbed topic
$\mathcal{T} = \langle \mathbb{T}_1, \cdots, \mathbb{T}_t \rangle$	- represents the true topic model
$\mathcal{T}' = \langle \mathbb{T}'_1, \cdots, \mathbb{T}'_t \rangle$	- represents the perturbed topic model
d	- chosen distance parameter
n	- total number of documents
D	- represents a document
$\mathcal{D} = \cup_{i=1}^{d} D$	- the corpus of documents
ω_j	- total number of unique words in document D_j
$\lvert w_{l,j} \rvert$	- count of word w_l in document D_j
$\lvert D_j \rvert$	- total number of words in the document D_j
$\mathcal{P}$	- the topic distribution for D_j from topic model $\mathcal{T}$
$\mathcal{P}'$	- the topic distribution for D_j from topic model $\mathcal{T}'$
$\mathcal{Q}_i$	- the word distribution for topic T_i in $\mathcal{T}$
$\mathcal{Q}'_i$	- the word distribution for topic T'_i in $\mathcal{T}'$
$p_{i,j}$	- probability of topic $\mathcal{T}_i$ occurring in document D_j output
$q_{i,l}$	- probability of word w_l occurring in topic $\mathcal{T}_i$
$q'_{i,l}$	- probability of word w_l occurring in topic $\mathcal{T}'_i$
$p'_{i,j}$	- probability of topic $\mathcal{T}'_i$ occurring in document D_j output
$T(w)$	- probability of word w occurring in topic T
$T'(w)$	- probability of word w occurring in topic T'

Definition 7.3.1. The total variation distance between two probability distributions P and Q is defined as

$$\delta_{TV}(P, Q) = \sup_{A \in \mathcal{F}} |P(A) - Q(A)| \tag{7.1}$$

where $\mathcal{F}$ represents a sigma-algebra on the subset of the sample space Ω.

Theorem 7.3.1. *For a countable set,* Ω

$$\delta_{TV}(P, Q) = \frac{1}{2}||P - Q||_1 = \frac{1}{2} \sum_{\omega \in \Omega} |P(\omega) - Q(\omega)| \tag{7.2}$$

Assumption 1. If a topic p has nonzero probability of occurring in a document, then the topic must contribute at least one count for each word in it for the document.

Lemma 7.3.2. *From assumption 1,*

$$\min_{i,j}\{p_{i,j}\} \geqslant \frac{k}{D_{max}} \tag{7.3}$$

Proof. Restating assumption 1 we get,

$$\begin{gathered}
\forall i, \forall j, \forall l \; i \in [t], j \in [n], l \in [|\mathcal{V}|] \\
p_{i,j} \cdot q_{i,l} \geqslant \frac{1}{|D_j|} \\
\Rightarrow p_{i,j} \cdot q_{i,l} \geqslant \min_j \left\{ \frac{1}{|D_j|} \right\} \\
\Rightarrow p_{i,j} \cdot q_{i,l} \geqslant \frac{1}{|D_{max}|} \\
\Rightarrow p_{i,j} \cdot \min_l\{q_{i,l}\} \geqslant \frac{1}{|D_{max}|} \\
\Rightarrow p_{i,j} \geqslant \frac{k}{|D_{max}|} [\because \min_l\{q_{i,l}\} \leqslant \frac{1}{k}]
\end{gathered}$$

Thus $\min_i\{p_{i,j}\} \geqslant \frac{k}{|D_{max}|}$ □

Lemma 7.3.3. *The maximum possible value of* p_{max} *is* $1 - (t-1) \cdot \frac{k}{|D_{max}|}$.

Proof. According to the proof statement, the topic mixture for a document D_j is given by $\mathcal{P}_j = \langle \frac{k}{|D_{max}|}, \cdots, \frac{k}{|D_{max}|}, 1 - (t-1) \cdot \frac{k}{|D_{max}|} \rangle$. We will prove this by contradiction. Let there exist some $\bar{p} \in \mathcal{P}_j$ such that $\bar{p} > \frac{k}{|D_{max}|}$. Clearly this means that $\max_{i,j}\{p_{i,j}\} = 1 - (t-2) \cdot \frac{k}{|D_{max}|} - \bar{p} < 1 - (t-1) \cdot \frac{k}{|D_{max}|}$. Clearly from lemma 7.3.2 this concludes our proof. □

Theorem 7.3.4. *For any pair of topics* $(\mathbb{T}, \mathbb{T}') \in \mathcal{T} \times \mathcal{T}'$,

$$||\mathbb{T} - \mathbb{T}'||_1 \geqslant 2 \frac{1}{\left(1 - (t-1)\frac{k}{\max_j |D_j|}\right)} \left(\frac{\mathcal{C}_{min}}{t} - \frac{1}{2}\left(1 - t\frac{k}{\max_j |D_j||}\right)\right),$$

where $\mathcal{C}_{min} = \min_{j,l}\left\{ \frac{\nu \cdot (|D_j| - |w_{l,j}|\omega_j)}{|D_j| \cdot (|D_j| + \nu \cdot \omega_j}} \right\}$, $|D_j|$ *is the total number of words in* D_j, ω_j *is the total number of unique words,* ν *is the variance of the distribution* $Lp(\epsilon', \delta', d)$,

$v = pre^{(\epsilon' \cdot \eta_0)/d}\left(\frac{1}{(1-r)^2} + \frac{2r}{(1-r)^3}\right) + p\bar{f}r\left(e^{-(\epsilon' \cdot \eta_0)/d} - e^{(\epsilon' \eta_0)/d}\right) - \left(pe^{(\epsilon' \cdot \eta_0)/d}\frac{r}{(1-r)^2} + pf(e^{-(\epsilon' \cdot \eta_0)/d} - e^{(\epsilon' \eta_0)/d})\right)^2$, $\bar{f} = \frac{df}{dr}$, $f = r\left(\frac{1-r^d - d \cdot r^{d-1}(1-r)}{(1-r)^2}\right)$, $r = e^{-(\epsilon'/d)}$, $p = \frac{e^{\epsilon/d}-1}{e^{\epsilon/d}+1}$, $\eta_0 = -\frac{d \cdot \ln((e^{\epsilon'/d}+1)\delta')}{\epsilon} + d$, $\epsilon' = \ln(1 + \frac{1}{\beta}(e^{\epsilon} - 1))$, $\delta' = \beta\delta$ *and* $|w_{l,j}|$ *is the number of times the word* $w_l \in \mathcal{V}$ *appears in transcript* D_j.

Proof. For any word w_l in a document $D_j, l \in [|\mathcal{V}|], j \in [n]$,

$$\sum_{i=1}^{t} p_{i,j} q_{i,l} = \frac{|w_l|}{|D_j|} \tag{7.4}$$

$$\sum_{i=1}^{t} p'_{i,j} q'_{i,l} = \frac{|w_l| + v}{|D_j| + v \cdot \omega_j} \tag{7.5}$$

Now subtracting eq 7.4 from eq 7.5 we

$$\sum_{i=1}^{t} (p'_{i,j} \cdot q'_{i,l} - p_{i,j} \cdot q_{i,l}) = \frac{v \cdot (|D_j| - |w_l|\omega_j)}{|D_j| \cdot (|D_j| + v \cdot \omega_j)} \tag{7.6}$$

Let $\mathcal{C}_{j,l} = \frac{v \cdot (|D_j| - |w_l|\omega_j)}{|D_j| \cdot (|D_j| + v \cdot \omega_j)}$.

$$\sum_{i=1}^{t} |p'_{i,j} \cdot q'_{i,l} - p_{i,j} \cdot q_{i,l}| \geqslant \sum_{i=1}^{t} (p'_{i,j} \cdot q'_{i,l} - p_{i,j} \cdot q_{i,l}) = \mathcal{C}_{j,l} \tag{7.7}$$

Now, observe that $min\{max_i\{|p'_{i,j} \cdot q'_{i,l} - p_{i,j} \cdot q_{i,l}|\}\}$ occurs when $|p'_{1,j} \cdot q'_{1,l} - p_{1,j} \cdot q_{1,l}| = \cdots = |p'_{t,j} \cdot q'_{t,l} - p_{t,j} \cdot q_{t,l}| \geqslant \frac{\mathcal{C}_{j,l}}{t}$. Let $p \in \mathcal{P}_j, p' \in \mathcal{P}'_j, q \in \mathcal{Q}_i$ and $q' \in \mathcal{Q}'_i$ such that $\mathcal{P}, \mathcal{P}', \mathcal{Q}, \mathcal{Q}'$ corresponds to $min\{max_i\{|p'_{i,j} \cdot q'_{i,l} - p_{i,j} \cdot q_{i,l}|\}\}$. Now renaming as follows

$$q_1 = max\{q, q'\} \tag{7.8}$$

$$p_1 = \begin{cases} p & \text{if } q_1 = q; \\ p' & \text{otherwise.} \end{cases} \tag{7.9}$$

$$q_2 = min\{q, q'\} \tag{7.10}$$

$$p_2 = \begin{cases} p & \text{if } q_2 = q; \\ p' & \text{otherwise.} \end{cases} \tag{7.11}$$

We get,

$$|p_1 \cdot q_1 - p_2 \cdot q_2| \geqslant \frac{\mathcal{C}_j}{t}$$

$$\Rightarrow |(q_1 - q_2) \cdot p_1 + q_2 \cdot (p_1 - p_2)| \geqslant \frac{\mathcal{C}_j}{t}$$

$$\Rightarrow |(q_1 - q_2) \cdot p_1| + |q_2 \cdot (p_1 - p_2)| \geqslant \frac{\mathcal{C}_{j,l}}{t}$$

$$\Rightarrow |(q_1 - q_2) \cdot p_1| + q_2 \cdot \left(1 - t \cdot \frac{k}{|D_{max}|}\right) \geqslant \frac{\mathcal{C}_j}{t} [\because \text{From lemma 7.3.3}]$$

$$\Rightarrow |(q_1 - q_2) \cdot p_1| + \frac{1}{2} \cdot \left(1 - t \cdot \frac{k}{|D_{max}|}\right) \geqslant \frac{\mathcal{C}_{j,l}}{t} [\because \text{ By eq 7.10} q_2 < \frac{1}{2}]$$

$$\Rightarrow (q_1 - q_2) \cdot \left(1 - (t-1) \cdot \frac{k}{|D_{max}|}\right) \geqslant \frac{\mathcal{C}_{j,l}}{t} - \frac{1}{2} \cdot \left(1 - t \cdot \frac{k}{|D_{max}|}\right)$$

$$[\because \text{ By lemma 7.3.3}]$$

$$\Rightarrow q_1 - q_2 \geqslant \frac{1}{\left(1 - (t-1) \cdot \frac{k}{|D_{max}|}\right)} \cdot \left(\frac{\mathcal{C}_{j,l}}{t} - \frac{1}{2} \cdot \left(1 - t \cdot \frac{k}{|D_{max}|}\right)\right)$$

Now clearly

$$min_{j,l}\{C_{j,l}\} = max_j\{min_l\{\frac{\frac{2d}{\epsilon} \cdot (|w_l|\omega_j - |D_j|)}{|D_j| \cdot (|D_j| + \frac{2d \cdot \omega_j}{\epsilon})}\}\} \tag{7.12}$$

Let $\mathcal{C}_{min} = min_{j,l}\{\mathcal{C}_{j,l}\}$. Thus for any pair $(T, T') \in \mathcal{T} \times \mathcal{T}'$ we have

$$\delta_{TV}(T, T') \geqslant sup_{w \in \mathcal{V}} |T(w) - T'(w)|$$

$$\geqslant \frac{1}{\left(1 - (t-1) \cdot \frac{k}{|D_{max}|}\right)} \cdot \left(\frac{\mathcal{C}_{min}}{t} - \frac{1}{2} \cdot \left(1 - t \cdot \frac{k}{|D_{max}|}\right)\right) \tag{7.13}$$

Thus, we have

$$||T - T'||_1 \geqslant 2 \cdot \frac{1}{\left(1 - (t-1) \cdot \frac{k}{|D_{max}|}\right)} \cdot \left(\frac{\mathcal{C}_{min}}{t} - \frac{1}{2} \cdot \left(1 - t \cdot \frac{k}{|D_{max}|}\right)\right) \tag{7.14}$$

□

7.4 Sensitive Keywords Lists

Table 7.3 presents the sensitive keywords list for each of Preech's evaluation datasets. We discuss Preech's efficacy in scrubbing them and its effect on the transcription utility in Sec. 3.7.

Datasets	Facebook 1	Facebook 2	Facebook 3	Carpenter 1	Carpenter 2
keywords	South Carolina	Grassley	Obama	Miller	Wessler
	Harvey	Cambridge Analytica	Tillis	Smith	Massachusetts
	Sandberg	South Dakota	Harris	Stewart	Kennedy
	Palo Alto		Kogan	kogan	Carpenter
	Cheryl		Alexander	Greenwood	Graham
				Roberts	Kogan
				Breyer	

Table 7.3: Sensitive keywords lists for each dataset

7.5 NLP Generated Text

In this section, we show a piece of the NLP-generated text using the GPT2 language model. The generated text is used as a corpus for dummy segments generation, as explained in Sec. 3.4. The model's input is a few sentences from the Carpenter case.

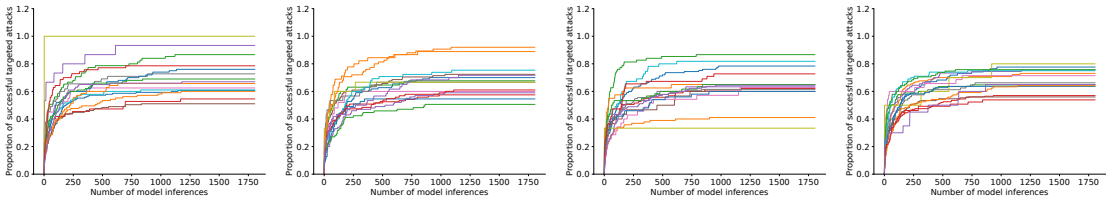
"In determining that the information is reasonable, the Court may find that this amount of intrusion is reasonable and that it is a reasonable effort by the government to harass privacy rights protections that normally apply to American citizens. Have there been any other issues before the Court before and after this hearing that we have not yet addressed. We will be looking again, on appeal. Would the government have done so differently today if they had arrested Petitioner five years earlier? The Court agrees that Petitioner has the right to petition the Court. I would agree with the court's conclusion that the government had previously attempted to arrest Petitioner, such as by conducting warrantless searches of his phone and computer in October 2009. There are some questions on the record, however, that we would have looked into in more detail in that regard.

DAVID GREENE: Well, they seem to have two thoughts that I am not going to sit down and define as true or false; I think – I've said it before. One thing we may know about this case is that the judge, Judith Miller, is a professor in Duke University's criminal-justice faculty and when she refused to reach a plea deal, the defense asked her to seek a final

decision in the case. They considered a reduced sentence because this is his plea deal, which means he's allowed to serve his entire life in prison."

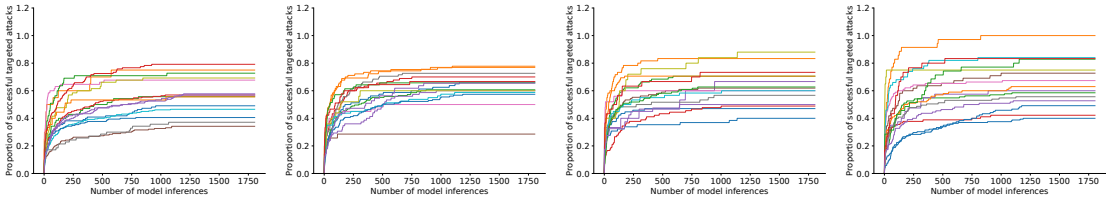
7.6 Differential Evolution Algorithm

To solve the optimization objective of Mystique efficiently, we ran differential evolution with *best2exp* strategy, with a population size of 100, maximum of 5 iterations, and tolerance of 0.001. Results for individual utterances are shown in Figs. 7.4 and 7.5.



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

Figure 7.4: Search performance over 1–4 different utterances.



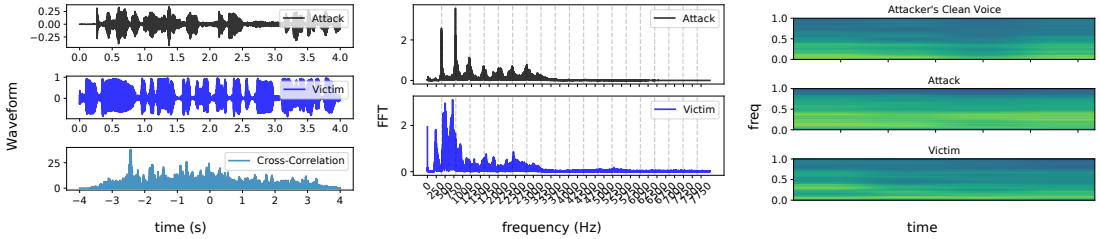
(a) (b) (c) (d)

Figure 7.5: Search performance over 5–8 different utterances.

7.7 Further Analysis of Mystique

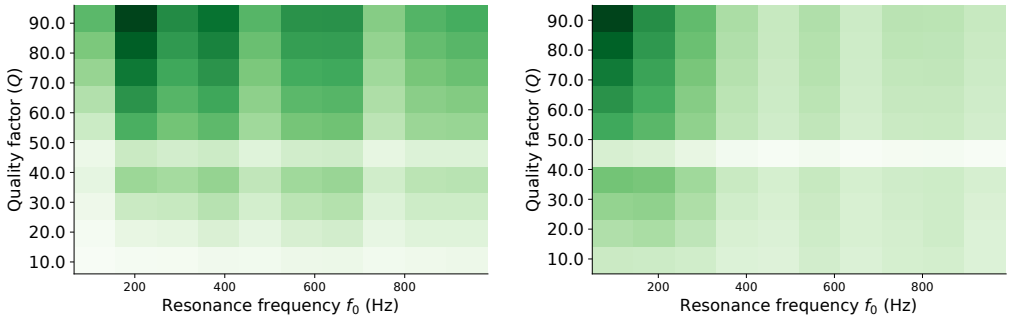
Model	Tubes						
	1	2	3	4	5	6	Avg
X-vector	87.53	85.8	82.45	76.47	84.22	85.95	83.74
SpeechBrain	88.88	84.31	83.69	82.56 2	80.38	85	84.14

Table 7.4: Mystique's over-the-air predictions consistency rate (%) across six repeated measurements.



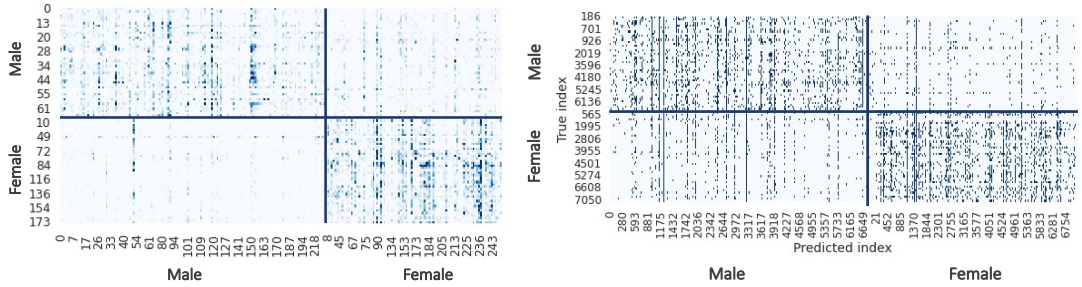
(a) Waveform (b) FFT (c) Spectrogram

Figure 7.6: Attack-victim pairs visualization when tube 1 (L = 40.6, d = 3.45 cm) is used: (a) the waveforms and their cross-correlation, (b) FFT, and (c) spectrogram for a deeper look at the spectral content. along with the FFT of the BPF model applied to the chirp signal.



(a) X-Vector (b) SpeechBrain

Figure 7.7: Successful impersonations histogram using a single-tube configuration on (a) x-vecotr and (b) SpeechBrain. Most of them are generated by tubes that have Low f_0 and high Q_0 values.



(a) X-Vector (b) SpeechBrain

Figure 7.8: The confusion matrix of (a) x-vector and (b) SpeechBrain's predictions on Mystique attack split by the true (attacker) and predicted (impersonated) speakers sex. The cross-sex submatrix is sparse, indicating attack is more successful within same-sex speakers.

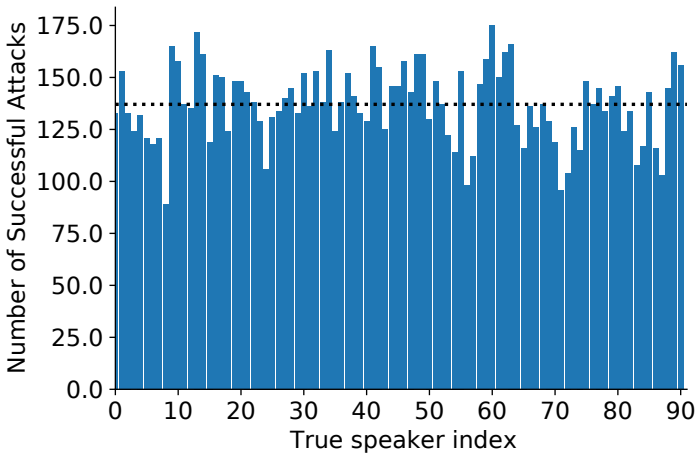


Figure 7.9: Number of successful impersonation attacks (out of 250) on the x-vector model for each adversarial speaker from our VoxCeleb test set.

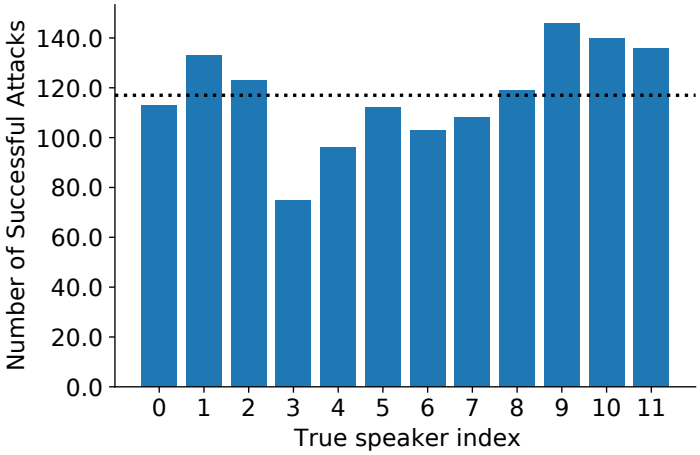
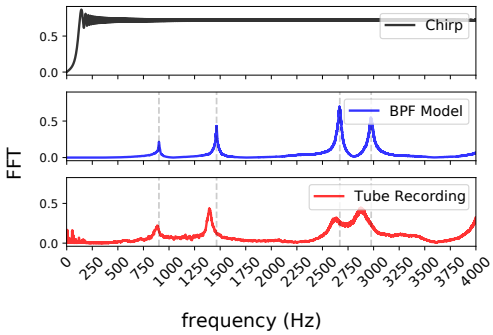


Figure 7.10: Number of successful attacks (false predictions) of the x-vector ASI model on the user study participants recordings.

(a) Two tubes connected with a HDF ring



(b) FFT of a chirp

Figure 7.11: Two-Tube structure and resonance effect.

Bibliography

[1] Anatomytool. "OpenStax AnatPhys fig.23.13 - The Esophagus - English labels" by OpenStax, license: CC BY. Source: book 'Anatomy and Physiology',https://openstax.org/details/books/anatomy-and-physiology.

[2] Lionel Sujay Vailshery. Smart speakers - statistics & facts. https://www.statista.com/topics/4748/smart-speakers/, 2021.

[3] Federica Laricchia. Number of digital voice assistants in use worldwide from 2019 to 2024. https://www.statista.com/statistics/973815/worldwide-digital-voice-assistant-in-use/, 2022.

[4] Anirudh Raju, Sankaran Panchapagesan, Xing Liu, Arindam Mandal, and Nikko Strom. Data augmentation for robust keyword spotting under playback interference. *arXiv preprint arXiv:1808.00563*, 2018.

[5] Dorian Lynskey. 'alexa, are you invading my privacy?' – the dark side of our voice assistants. https://tinyurl.com/y5velcwz, 2019.

[6] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. IEEE, 2013.

[7] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, and Yonghui Wu. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in Neural Information Processing Systems 31*, pages 4480–4490. Curran Associates, Inc., 2018.

[8] Björn Schuller, Stefan Steidl, Anton Batliner, Felix Burkhardt, Laurence Devillers, Christian MüLler, and Shrikanth Narayanan. Paralinguistics in speech and language—state-of-the-art and the challenge. *Computer Speech & Language*, 27(1):4–39, 2013.

[9] Andreas Nautsch, Catherine Jasserand, Els Kindt, Massimiliano Todisco, Isabel Trancoso, and Nicholas Evans. The gdpr & speech data: Reflections of legal and technology communities, first steps towards a common understanding. *arXiv preprint arXiv:1907.03458*, 2019.

[10] Margaret Davino. Assessing privacy risk in outsourcing. *Assessing Privacy Risk in Outsourcing/AHIMA, American Health Information Management Association*, 2004.

[11] Voice ID - Customer Service - HSBC Bank USA — us.hsbc.com. `https://www.us.hsbc.com/customer-service/voice/`. [Accessed 04-01-2024].

[12] Voice ID | chase.com — chase.com. `https://www.chase.com/personal/voice-biometrics`. [Accessed 04-01-2024].

[13] P Jonathon Phillips, Patrick Grother, Ross Micheals, Duane M Blackburn, Elham Tabassi, and Mike Bone. Face recognition vendor test 2002. In *2003 IEEE International SOI Conference. Proceedings (Cat. No. 03CH37443)*, page 44. IEEE, 2003.

[14] Pawel Drozdowski, Christian Rathgeb, Antitza Dantcheva, Naser Damer, and Christoph Busch. Demographic bias in biometrics: A survey on an emerging challenge. *IEEE Transactions on Technology and Society*, 1(2):89–103, 2020.

[15] Salem Hamed Abdurrahim, Salina Abdul Samad, and Aqilah Baseri Huddin. Review on the effects of age, gender, and race demographics on automatic face recognition. *The Visual Computer*, 34:1617–1630, 2018.

[16] Brendan F Klare, Mark J Burge, Joshua C Klontz, Richard W Vorder Bruegge, and Anil K Jain. Face recognition performance: Role of demographic information. *IEEE Transactions on information forensics and security*, 7(6):1789–1801, 2012.

[17] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR, 2018.

[18] Vitor Albiero, Krishnapriya Ks, Kushal Vangara, Kai Zhang, Michael C King, and Kevin W Bowyer. Analysis of gender inequality in face recognition accuracy. In *Proceedings of the ieee/cvf winter conference on applications of computer vision workshops*, pages 81–89, 2020.

[19] Vítor Albiero, Kevin Bowyer, Kushal Vangara, and Michael King. Does face recognition accuracy get better with age? deep face matchers say no. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 261–269, 2020.

[20] Boyu Lu, Jun-Cheng Chen, Carlos D Castillo, and Rama Chellappa. An experimental evaluation of covariates effects on unconstrained face verification. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 1(1):42–55, 2019.

[21] Philippe G Schyns and Heinrich H Bulthoff. Viewpoint dependence and face recognition. In *Proceedings of the Sixteenth Annual Conference of the Cognitive Science Society*, pages 789–793. Routledge, 2019.

[22] Aman Bhatta, Vítor Albiero, Kevin W Bowyer, and Michael C King. The gender gap in face recognition accuracy is a hairy problem. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 303–312, 2023.

[23] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and Simile Classifiers for Face Verification. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2009.

[24] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014.

[25] Omkar Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC 2015-Proceedings of the British Machine Vision Conference 2015*. British Machine Vision Association, 2015.

[26] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[27] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[28] Iacopo Masi, Yue Wu, Tal Hassner, and Prem Natarajan. Deep face recognition: A survey. In *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pages 471–478. IEEE, 2018.

[29] Isabelle Hupont and Carles Fernández. Demogpairs: Quantifying the impact of demographic imbalance in deep face recognition. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–7, 2019.

[30] Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1548–1558, 2021.

[31] Joseph P Robinson, Can Qin, Yann Henon, Samson Timoner, and Yun Fu. Balancing biases and preserving privacy on balanced faces in the wild. *IEEE Transactions on Image Processing*, 2023.

[32] Haiyu Wu and Kevin W Bowyer. A real balanced dataset for understanding bias? factors that impact accuracy, not numbers of identities and images. *arXiv preprint arXiv:2304.09818*, 2023.

[33] Shimaa Ahmed, Ilia Shumailov, Nicolas Papernot, and Kassem Fawaz. Towards more robust keyword spotting for voice assistants. In *31st USENIX Security Symposium*, 2022.

[34] Shimaa Ahmed, Amrita Roy Chowdhury, Kassem Fawaz, and Parmesh Ramanathan. Preech: A system for {Privacy-Preserving} speech transcription. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2703–2720, 2020.

[35] Shimaa Ahmed, Yash Wani, Ali Shahin Shamsabadi, Mohammad Yaghini, Ilia Shumailov, Nicolas Papernot, and Kassem Fawaz. Tubes among us: Analog attack on automatic speaker identification. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 265–282, 2023.

[36] Harrison Rosenberg, Shimaa Ahmed, Guruprasad V Ramesh, Ramya Korlakai Vinayak, and Kassem Fawaz. Unbiased face synthesis with diffusion models: Are we there yet? *arXiv preprint arXiv:2309.07277*, 2023.

[37] Lea Schönherr, Maximilian Golla, Thorsten Eisenhofer, Jan Wiele, Dorothea Kolossa, and Thorsten Holz. Unacceptable, where is my privacy? exploring accidental triggers of smart speakers. *arXiv preprint arXiv:2008.00508*, 2020.

[38] Guoguo Chen, Carolina Parada, and Georg Heigold. Small-footprint keyword spotting using deep neural networks. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4087–4091. IEEE, 2014.

[39] Siddharth Sigtia, Rob Haynes, Hywel Richards, Erik Marchi, and John Bridle. Efficient voice trigger detection for low resource hardware. In *Interspeech*, pages 2092–2096, 2018.

[40] Assaf Hurwitz Michaely, Xuedong Zhang, Gabor Simko, Carolina Parada, and Petar Aleksic. Keyword spotting for google assistant using contextual speech recognition. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 272–278. IEEE, 2017.

[41] Yuxuan Chen, Xuejing Yuan, Jiangshan Zhang, Yue Zhao, Shengzhi Zhang, Kai Chen, and XiaoFeng Wang. Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices. In *29th USENIX Security Symposium (USENIX Security 20)*, 2020.

[42] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. In *International conference on machine learning*, pages 5231–5240. PMLR, 2019.

[43] Juncheng Li, Shuhui Qu, Xinjian Li, Joseph Szurley, J Zico Kolter, and Florian Metze. Adversarial music: Real world audio adversary against wake-word detection system. In *Advances in Neural Information Processing Systems*, pages 11908–11918, 2019.

[44] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. 2018.

[45] Muhammad Ejaz Ahmed, Il-Youp Kwak, Jun Ho Huh, Iljoo Kim, Taekkyung Oh, and Hyoungshick Kim. Void: A fast and light voice liveness detection system. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 2685–2702, 2020.

[46] Florian Tramer, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*, 2020.

[47] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, 2017.

[48] Daniel J Dubois, Roman Kolcun, Anna Maria Mandalari, Muhammad Talha Paracha, David Choffnes, and Hamed Haddadi. When speakers are all ears: Characterizing misactivations of iot smart speakers. *Proceedings on Privacy Enhancing Technologies*, 2020(4):255–276, 2020.

[49] Leena Mary and G Deekshitha. *Searching Speech Databases: Features, Techniques and Evaluation Measures*. Springer, 2018.

[50] Tara N Sainath and Carolina Parada. Convolutional neural networks for small-footprint keyword spotting. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[51] Raphael Tang and Jimmy Lin. Deep residual learning for small-footprint keyword spotting. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5484–5488. IEEE, 2018.

[52] Seungwoo Choi, Seokjun Seo, Beomjun Shin, Hyeongmin Byun, Martin Kersner, Beomsu Kim, Dongyoung Kim, and Sungjoo Ha. Temporal convolution for real-time keyword spotting on mobile devices. *arXiv preprint arXiv:1904.03814*, 2019.

[53] Noura Abdi, Kopo M Ramokapane, and Jose M Such. More than smart speakers: security and privacy perceptions of smart home personal assistants. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*, 2019.

[54] Varun Chandrasekaran, Suman Banerjee, Bilge Mutlu, and Kassem Fawaz. Powercut and obfuscator: An exploration of the design space for privacy-preserving interventions for smart speakers. In *Seventeenth Symposium on Usable Privacy and Security ({SOUPS} 2021)*, pages 535–552, 2021.

[55] Nathan Malkin, Joe Deatrick, Allen Tong, Primal Wijesekera, Serge Egelman, and David Wagner. Privacy attitudes of smart speaker users. *Proceedings on Privacy Enhancing Technologies*, 2019(4):250–271, 2019.

[56] Eric Zeng, Shrirang Mare, and Franziska Roesner. End user security and privacy concerns with smart homes. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 65–80, 2017.

[57] Jide S Edu, Jose M Such, and Guillermo Suarez-Tangil. Smart home personal assistants: a security and privacy review. *arXiv preprint arXiv:1903.05593*, 2019.

[58] Joseph Bugeja, Andreas Jacobsson, and Paul Davidsson. On privacy and security challenges in smart connected homes. In *2016 European Intelligence and Security Informatics Conference (EISIC)*, pages 172–175. IEEE, 2016.

[59] Ilia Shumailov, Yiren Zhao, Daniel Bates, Nicolas Papernot, Robert Mullins, and Ross Anderson. Sponge examples: Energy-latency attacks on neural networks. In *6th IEEE European Symposium on Security and Privacy (EuroS&P)*, 2021.

[60] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[61] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[62] Lea Schönherr, Thorsten Eisenhofer, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Imperio: Robust over-the-air adversarial examples for automatic speech recognition systems. *arXiv preprint arXiv:1908.01551*, 2019.

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

[64] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[65] Yuantian Miao, Ben Zi Hao Zhao, Minhui Xue, Chao Chen, Lei Pan, Jun Zhang, Dali Kaafar, and Yang Xiang. The audio auditor: Participant-level membership inference in voice-based iot. *arXiv preprint arXiv:1905.07082*, 2019.

[66] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. Updates-leak: Data set inference and reconstruction attacks in online learning, 2019.

[67] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks, 2019.

[68] Almos Zarandy, Ilia Shumailov, and Ross Anderson. Hey alexa what did i just type? decoding smartphone sounds with a voice assistant, 2020.

[69] Miles Brignall. Amazon hit with major data breach days before black friday. `https://tinyurl.com/yyu4mmj8`, 2018.

[70] Joseph Cox. Amazon fired employee for leaking customer emails. `https://tinyurl.com/hvv36tcj`, 2020.

[71] Matt Day, Giles Turner, and Natalia Drozdiak. Amazon workers are listening to what you tell alexa. `https://tinyurl.com/audw32jw`, 2019.

[72] Tim Verheyden, Denny Baert, Lente Van Hee, and Ruben Van Den Heuvel. Google employees are eavesdropping, even in your living room, vrt nws has discovered, 2019.

[73] Chavie Lieber. Amazon's alexa might be a key witness in a murder case. `https://tinyurl.com/4v6te95z`, 2018.

[74] Nirupam Roy, Haitham Hassanieh, and Romit Roy Choudhury. Backdoor: Making microphones hear inaudible sounds. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pages 2–14, 2017.

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

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

[77] Sourav Bhattacharya, Dionysis Manousakas, Alberto Gil CP Ramos, Stylianos I Venieris, Nicholas D Lane, and Cecilia Mascolo. Countering acoustic adversarial attacks in microphone-equipped smart home devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):1–24, 2020.

[78] Chen Wang, S Abhishek Anand, Jian Liu, Payton Walker, Yingying Chen, and Nitesh Saxena. Defeating hidden audio channel attacks on voice assistants via audio-induced surface vibrations. In *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC '19, page 42–56, New York, NY, USA, 2019. Association for Computing Machinery.

[79] Stephen B Vardeman and Max D Morris. Majority voting by independent classifiers can increase error rates. *The American Statistician*, 67(2):94–96, 2013.

[80] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[81] Jean-Marc Jot, Laurent Cerveau, and Olivier Warusfel. Analysis and synthesis of room reverberation based on a statistical time-frequency model. In *Audio Engineering Society Convention 103*. Audio Engineering Society, 1997.

[82] Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. The faults in our asrs: An overview of attacks against automatic speech recognition and speaker identification systems. *IEEE Symposium on Security and Privacy*, 2021.

[83] Pete Warden. Launching the speech commands dataset. Available: `https://tinyurl.com/yy4sa92z`, August, 2017.

[84] Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 351–355. IEEE, 2018.

[85] Jont B Allen and David A Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

[86] Gildas Avoine, Muhammed Ali Bingöl, Ioana Boureanu, Srdjan Čapkun, Gerhard Hancke, Süleyman Kardaş, Chong Hee Kim, Cédric Lauradoux, Benjamin Martin, Jorge Munilla, et al. Security of distance-bounding: A survey. *ACM Computing Surveys (CSUR)*, 51(5):1–33, 2018.

[87] S. Mauw, Z. Smith, J. Toro-Pozo, and R. Trujillo-Rasua. Distance-bounding protocols: Verification without time and location. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 549–566, 2018.

[88] Y. Gong and C. Poellabauer. Protecting voice controlled systems using sound source identification based on acoustic cues. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9, 2018.

[89] Ry Crist and Andrew Gebhart. Everything you need to know about the amazon echo. `https://tinyurl.com/r2ura464`, 2018.

[90] Ravie Lakshmanan. Apple and google suspend monitoring of voice recordings by humans (update: Amazon too). `https://tinyurl.com/y34hjv3c`, 2019.

[91] Abner Li. 'hey google' sensitivity setting now official, gradually rolling out. `https://9to5google.com/2020/04/23/hey-google-sensitivity/`, 2020.

[92] Alex Hern. Smart speaker statistics. `https://tinyurl.com/3nnu6n84`, 2021.

[93] Bjørn Karmann. Project alias. `https://bjoernkarmann.dk/project_alias`, 2018.

[94] Pingping Wu, Hong Liu, Xiaofei Li, Ting Fan, and Xuewu Zhang. A novel lip descriptor for audio-visual keyword spotting based on adaptive decision fusion. *IEEE Transactions on Multimedia*, 18(3):326–338, 2016.

[95] Abraham Mhaidli, Manikandan Kandadai Venkatesh, Yixin Zou, and Florian Schaub. Listen only when spoken to: Interpersonal communication cues as smart speaker privacy controls. *Proceedings on Privacy Enhancing Technologies*, 2020(2):251–270, 2020.

[96] Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, et al. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*, 2018.

[97] Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Advances in Neural Information Processing Systems*, pages 11292–11303, 2019.

[98] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[99] Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *IEEE S&P 2019*, 2018.

[100] Xinyu Lei, Guan-Hua Tu, Alex X. Liu, Chi-Yu Li, and Tian Xie. The insecurity of home digital voice assistants - amazon alexa as a case study. *CoRR*, abs/1712.03327, 2017.

[101] S. Chen, K. Ren, S. Piao, C. Wang, Q. Wang, J. Weng, L. Su, and A. Mohaisen. 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)*, pages 183–195, 2017.

[102] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182, 2016.

[103] Susan E McGregor, Polina Charters, Tobin Holliday, and Franziska Roesner. Investigating the computer security practices and needs of journalists. In *24th {USENIX} Security Symposium ({USENIX} Security 15)*, pages 399–414, 2015.

[104] Saeid Safavi, Martin Russell, and Peter Jančovič. Automatic speaker, age-group and gender identification from children's speech. *Computer Speech & Language*, 50:141–156, 2018.

[105] Björn Schuller and Anton Batliner. *Computational paralinguistics: emotion, affect and personality in speech and language processing*. John Wiley & Sons, 2013.

[106] Ranya Aloufi, Hamed Haddadi, and David Boyle. Emotionless: Privacy-preserving speech analysis for voice assistants. *arXiv preprint arXiv:1908.03632*, 2019.

[107] Hannah Muckenhirn, Mathew Magimai Doss, and Sébastien Marcell. Towards directly modeling raw speech signal for speaker verification using cnns. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4884–4888. IEEE, 2018.

[108] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[109] Tavish Vaidya and Micah Sherr. You talk too much: Limiting privacy exposure via voice input. In *International Workshop on Privacy Engineering (IWPE)*, 2019.

[110] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[111] An all-neural on-device speech recognizer. `https://ai.googleblog.com/2019/03/an-all-neural-on-device-speech.html`.

[112] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1. *NASA STI/Recon technical report n*, 93, 1993.

[113] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an ASR corpus based on public domain audio books. In *ICASSP 2015*, pages 5206–5210. IEEE, 2015.

[114] Gautham J Mysore. Can we automatically transform speech recorded on common consumer devices in real-world environments into professional production quality speech?—a dataset, insights, and challenges. *IEEE Signal Processing Letters*, 22(8):1006–1010, 2014.

[115] Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, Tomi Kinnunen, and Zhenhua Ling. The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods. *arXiv preprint arXiv:1804.04262*, 2018.

[116] Christophe Veaux, Junichi Yamagishi, Kirsten MacDonald, et al. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. *University of Edinburgh. The Centre for Speech Technology Research (CSTR)*, 2017.

[117] Zhizheng Wu, Nicholas Evans, Tomi Kinnunen, Junichi Yamagishi, Federico Alegre, and Haizhou Li. Spoofing and countermeasures for speaker verification. *Speech Commun.*, 66(C):130–153, February 2015.

[118] Johan Lindberg and Mats Blomberg. Vulnerability in speaker verification-a study of technical impostor techniques. In *Sixth European Conference on Speech Communication and Technology*, 1999.

[119] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005.

[120] Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[121] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *EMNLP 2009*, pages 248–256, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[122] Sarana Nutanong, Chenyun Yu, Raheem Sarwar, Peter Xu, and Dickson Chow. A scalable framework for stylometric analysis query processing. In *ICDM 2016*, pages 1125–1130. IEEE, 2016.

[123] Kazuhiro Kobayashi and Tomoki Toda. sprocket: Open-source voice conversion software. In *Odyssey*, pages 203–210, 2018.

[124] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng. Phonetic posteriorgrams for many-to-one voice conversion without parallel data training. In *ICME 2016*, pages 1–6. IEEE, 2016.

[125] K Sri Rama Murty, Bayya Yegnanarayana, and M Anand Joseph. Characterization of glottal activity from speech signals. *IEEE signal processing letters*, 16(6):469–472, 2009.

[126] Paul Boersma. Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Institute of Phonetic Sciences - University of Amsterdam*, 17:97–110, 1993.

[127] Rui Chen, Noman Mohammed, Benjamin CM Fung, Bipin C Desai, and Li Xiong. Publishing set-valued data via differential privacy. *Proceedings of the VLDB Endowment*, 4(11):1087–1098, 2011.

[128] Arik Friedman and Assaf Schuster. Data mining with differential privacy. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 493–502. ACM, 2010.

[129] Johes Bater, Xi He, William Ehrich, Ashwin Machanavajjhala, and Jennie Rogers. Shrinkwrap: Differentially-private query processing in private data federations. *arXiv preprint arXiv:1810.01816*, 2018.

[130] Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. Broadening the scope of differential privacy using metrics. In Emiliano De Cristofaro and Matthew Wright, editors, *Privacy Enhancing Technologies*, pages 82–102, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[131] Thomas Hofmann. Probabilistic latent semantic analysis. *arXiv preprint arXiv:1301.6705*, 2013.

[132] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[133] Anthony Rousseau, Paul Deléglise, and Yannick Esteve. Ted-lium: an automatic speech recognition dedicated corpus. In *LREC*, pages 125–129, 2012.

[134] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[135] Brij Mohan Lal Srivastava, Aurélien Bellet, Marc Tommasi, and Emmanuel Vincent. Privacy-Preserving Adversarial Representation Learning in ASR: Reality or Illusion? In *INTERSPEECH 2019*, Graz, Austria, September 2019.

[136] Manas A Pathak, Bhiksha Raj, Shantanu D Rane, and Paris Smaragdis. Privacy-preserving speech processing: cryptographic and string-matching frameworks show promise. *IEEE signal processing magazine*, 30(2):62–74, 2013.

[137] Ferdinand Brasser, Tommaso Frassetto, Korbinian Riedhammer, Ahmad-Reza Sadeghi, Thomas Schneider, and Christian Weinert. Voiceguard: Secure and private speech processing. In *Interspeech*, pages 1303–1307, 2018.

[138] Jianwei Qian, Haohua Du, Jiahui Hou, Linlin Chen, Taeho Jung, Xiang-Yang Li, Yu Wang, and Yanbo Deng. Voicemask: Anonymize and sanitize voice input on mobile devices. *arXiv preprint arXiv:1711.11460*, 2017.

[139] Jianwei Qian, Feng Han, Jiahui Hou, Chunhong Zhang, Yu Wang, and Xiang-Yang Li. Towards privacy-preserving speech data publishing. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1079–1087. IEEE, 2018.

[140] Wilson Cai, Anish Doshi, and Rafael Valle. Attacking speaker recognition with deep generative models. *arXiv preprint arXiv:1801.02384*, 2018.

[141] Felix Kreuk, Yossi Adi, Moustapha Cisse, and Joseph Keshet. Fooling end-to-end speaker verification with adversarial examples. *ICASSP 2018*, Apr 2018.

[142] A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017.

[143] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, April 2018.

[144] Paul Taylor. *Text-to-speech synthesis*. Cambridge university press, 2009.

[145] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016.

[146] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

[147] Lindasalwa Muda, Mumtaj Begam, and I. Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques, 2010.

[148] Berrak Sisman, Mingyang Zhang, Sakriani Sakti, Haizhou Li, and Satoshi Nakamura. Adaptive wavenet vocoder for residual compensation in gan-based voice conversion. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 282–289, 2018.

[149] Siyang Yuan, Pengyu Cheng, Ruiyi Zhang, Weituo Hao, Zhe Gan, and Lawrence Carin. Improving zero-shot voice style transfer via disentangled representation learning, 2021.

[150] J. Lindberg and M. Blomberg. Vulnerability in speaker verification - a study of technical impostor techniques, 1999.

[151] Songxiang Liu, Haibin Wu, Hung yi Lee, and Helen Meng. Adversarial attacks on spoofing countermeasures of automatic speaker verification, 2019.

[152] Andre Kassis and Urs Hengartner. Practical attacks on voice spoofing countermeasures, 2021.

[153] Ting Chen, Hongwei Luo, Yijie Shen, Feng Lin, and Guoai Xu. Spoofing speaker verification system by adversarial examples leveraging the generalized speaker difference. *Security and Communication Networks*, 2021:6664578, 2021.

[154] Jiacheng Shang, Si Chen, and Jie Wu. Defending against voice spoofing: A robust software-based liveness detection system. In *2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 28–36, 2018.

[155] Yao Wang, Wandong Cai, Tao Gu, Wei Shao, Yannan Li, and Yong Yu. Secure your voice: An oral airflow-based continuous liveness detection for voice assistants. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(4), December 2019.

[156] Agnieszka Owczarek and Krzysztof Ślot. Lipreading procedure for liveness verification in video authentication systems. In Emilio Corchado, Václav Snášel, Ajith Abraham, Michał Woźniak, Manuel Graña, and Sung-Bae Cho, editors, *Hybrid Artificial Intelligent Systems*, pages 115–124, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[157] Linghan Zhang, Sheng Tan, and Jie Yang. 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*, CCS '17, page 57–71, New York, NY, USA, 2017. Association for Computing Machinery.

[158] Lawrence George Kersta. Voiceprint identification. *The Journal of the Acoustical Society of America*, 34(5):725–725, 1962.

[159] Yan Meng, Zichang Wang, Wei Zhang, Peilin Wu, Haojin Zhu, Xiaohui Liang, and Yao Liu. Wivo: Enhancing the security of voice control system via wireless signal in iot environment. In *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Mobihoc '18, page 81–90, New York, NY, USA, 2018. Association for Computing Machinery.

[160] Andrew Allen and Nikunj Raghuvanshi. Aerophones in flatland: Interactive wave simulation of wind instruments. *ACM Transactions on Graphics (TOG)*, 34(4):1–11, 2015.

[161] Nobuyuki Umetani, Athina Panotopoulou, Ryan Schmidt, and Emily Whiting. Printone: interactive resonance simulation for free-form print-wind instrument design. *ACM Transactions on Graphics (TOG)*, 35(6):1–14, 2016.

[162] Lawrence E Kinsler, Austin R Frey, Alan B Coppens, and James V Sanders. *Fundamentals of acoustics*. John wiley & sons, 2000.

[163] Abdulaziz M Aljalal. Sound resonance in pipes with discrete fourier transform. *European Journal of Physics*, 36(5):055030, aug 2015.

[164] Cornelis Johannes Nederveen. Acoustical aspects of woodwind instruments. 1969.

[165] AE Bate. Lx.(i.) the end-corrections of an open organ flue-pipe; and (ii.) the acoustical conductance of orifices. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 10(65):617–632, 1930.

[166] Michael J Moloney and Daniel L Hatten. Acoustic quality factor and energy losses in cylindrical pipes. *American Journal of Physics*, 69(3):311–314, 2001.

[167] Johan Liljencrants. Tubes quality factor. `http://www.fonema.se/qpipe/qpipe.htm`.

[168] K Stevens. Acoustic phonetics, cambridge, 1998.

[169] John R. Deller, John H. L. Hansen, and John G. Proakis. *Speech Production and Modeling*, pages 97–97. Wiley-IEEE Press, 2000.

[170] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur. Deep neural network embeddings for text-independent speaker verification. In *Interspeech*, pages 999–1003, 2017.

[171] Kenneth N Stevens. *Acoustic phonetics*, volume 30. MIT press, 2000.

[172] Tamara Smyth and Jonathan S Abel. Estimating waveguide model elements from acoustic tube measurements. *Acta Acustica united with Acustica*, 95(6):1093–1103, 2009.

[173] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5220–5224. IEEE, 2017.

[174] Ayesha Pervaiz, Fawad Hussain, Huma Israr, Muhammad Ali Tahir, Fawad Riasat Raja, Naveed Khan Baloch, Farruh Ishmanov, and Yousaf Bin Zikria. Incorporating noise robustness in speech command recognition by noise augmentation of training data. *Sensors*, 20(8):2326, 2020.

[175] Hu Hu, Tian Tan, and Yanmin Qian. Generative adversarial networks based data augmentation for noise robust speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5044–5048. IEEE, 2018.

[176] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset shift in machine learning*. Mit Press, 2008.

[177] Masashi Sugiyama and Motoaki Kawanabe. *Machine learning in non-stationary environments: Introduction to covariate shift adaptation*. MIT press, 2012.

[178] R.J. McAulay and T.F. Quatieri. Pitch estimation and voicing detection based on a sinusoidal speech model. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 249–252 vol.1, 1990.

[179] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[180] Jong Wook Kim, Justin Salamon, Peter Li, and Juan Pablo Bello. Crepe: A convolutional representation for pitch estimation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 161–165, 2018.

[181] Rainer Storn and Kenneth Price. Differential evolution –a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[182] Ali Shahin Shamsabadi, Francisco Sepúlveda Teixeira, Alberto Abad, Bhiksha Raj, Andrea Cavallaro, and Isabel Trancoso. Foolhd: Fooling speaker identification by highly imperceptible adversarial disturbances. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6159–6163. IEEE, 2021.

[183] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *arXiv preprint arXiv:2005.07143*, 2020.

[184] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Science and Language*, 2019.

[185] J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018.

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

[187] Emily Wenger, Max Bronckers, Christian Cianfarani, Jenna Cryan, Angela Sha, Haitao Zheng, and Ben Y Zhao. " hello, it's me": Deep learning-based speech synthesis attacks in the real world. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 235–251, 2021.

[188] Galina Lavrentyeva, Sergey Novoselov, Egor Malykh, Alexander Kozlov, Oleg Kudashev, and Vadim Shchemelinin. Audio replay attack detection with deep learning frameworks. In *Interspeech*, pages 82–86, 2017.

[189] Stephan Rabanser, Anvith Thudi, Kimia Hamidieh, Adam Dziedzic, and Nicolas Papernot. Selective classification via neural network training dynamics. *arXiv preprint arXiv:2205.13532*, 2022.

[190] Nadine Lavan, Sophie K Scott, and Carolyn McGettigan. Impaired generalization of speaker identity in the perception of familiar and unfamiliar voices. *Journal of Experimental Psychology: General*, 145(12):1604, 2016.

[191] Stephen J Winters, Susannah V Levi, and David B Pisoni. Identification and discrimination of bilingual talkers across languages. *The Journal of the Acoustical Society of America*, 123(6):4524–4538, 2008.

[192] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, et al. Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection. *arXiv preprint arXiv:2109.00537*, 2021.

[193] Alessandro Pianese, Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Deepfake audio detection by speaker verification. *arXiv preprint arXiv:2209.14098*, 2022.

[194] Nicolas M Müller, Pavel Czempin, Franziska Dieckmann, Adam Froghyar, and Konstantin Böttinger. Does audio deepfake detection generalize? *arXiv preprint arXiv:2203.16263*, 2022.

[195] Stefano Borzì, Oliver Giudice, Filippo Stanco, and Dario Allegra. Is synthetic voice detection research going into the right direction? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 71–80, 2022.

[196] Logan Blue, Luis Vargas, and Patrick Traynor. 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*, WiSec '18, page 123–133, New York, NY, USA, 2018. Association for Computing Machinery.

[197] Chen Yan, Yan Long, Xiaoyu Ji, and Wenyuan Xu. 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*, CCS '19, page 1215–1229, New York, NY, USA, 2019. Association for Computing Machinery.

[198] Linghan Zhang, Sheng Tan, Jie Yang, and Yingying Chen. 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*, CCS '16, page 1080–1091, New York, NY, USA, 2016. Association for Computing Machinery.

[199] Logan Blue, Hadi Abdullah, Luis Vargas, and Patrick Traynor. 2ma: Verifying voice commands via two microphone authentication. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, ASIACCS '18, page 89–100, New York, NY, USA, 2018. Association for Computing Machinery.

[200] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning, 2017.

[201] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. A general framework for adversarial examples with objectives. *ACM Transactions on Privacy and Security*, 22(3):1–30, Jul 2019.

[202] Zuxuan Wu, Ser-Nam Lim, Larry Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors, 2020.

[203] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world, 2020.

[204] Stepan Komkov and Aleksandr Petiushko. Advhat: Real-world adversarial attack on arcface face id system. *2020 25th International Conference on Pattern Recognition (ICPR)*, Jan 2021.

[205] Zhe Zhou, Di Tang, Xiaofeng Wang, Weili Han, Xiangyu Liu, and Kehuan Zhang. Invisible mask: Practical attacks on face recognition with infrared, 2018.

[206] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection, 2019.

[207] Lauren Feiner and Annie Palmer. Rules around facial recognition and policing remain blurry. *CNBC Tech*, 2021.

[208] Antoaneta Roussi. Resisting the rise of facial recognition. *Nature*, 2020.

[209] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[210] Mandar Dixit, Roland Kwitt, Marc Niethammer, and Nuno Vasconcelos. Aga: Attribute-guided augmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7455–7463, 2017.

[211] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. In *ICLR 2023 Workshop on Mathematical and Empirical Understanding of Foundation Models*, 2023.

[212] Brandon Smith, Miguel Farinha, Siobhan Mackenzie Hall, Hannah Rose Kirk, Aleksandar Shtedritski, and Max Bain. Balancing the picture: Debiasing vision-language datasets with synthetic contrast sets. *arXiv preprint arXiv:2305.15407*, 2023.

[213] Felix Friedrich, Patrick Schramowski, Manuel Brack, Lukas Struppek, Dominik Hintersdorf, Sasha Luccioni, and Kristian Kersting. Fair diffusion: Instructing text-to-image generation models on fairness. *arXiv preprint arXiv:2302.10893*, 2023.

[214] Linoy Tsaban and Apolinário Passos. Ledits: Real image editing with ddpm inversion and semantic guidance, 2023.

[215] Manuel Brack, Felix Friedrich, Dominik Hintersdorf, Lukas Struppek, Patrick Schramowski, and Kristian Kersting. Sega: Instructing diffusion using semantic dimensions. *arXiv preprint arXiv:2301.12247*, 2023.

[216] Malsha V Perera and Vishal M Patel. Analyzing bias in diffusion-based face generation models. *arXiv preprint arXiv:2305.06402*, 2023.

[217] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[218] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023.

[219] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023.

[220] Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing, 2023.

[221] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[222] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[223] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[224] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[225] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6007–6017, 2023.

[226] Preethi Seshadri, Sameer Singh, and Yanai Elazar. The bias amplification paradox in text-to-image generation. *arXiv preprint arXiv:2308.00755*, 2023.

[227] Alexandra Sasha Luccioni, Christopher Akiki, Margaret Mitchell, and Yacine Jernite. Stable bias: Analyzing societal representations in diffusion models. *arXiv preprint arXiv:2303.11408*, 2023.

[228] Lukas Struppek, Dominik Hintersdorf, and Kristian Kersting. The biased artist: Exploiting cultural biases via homoglyphs in text-guided image generation models. *arXiv preprint arXiv:2209.08891*, 2022.

[229] Cristian Muñoz, Sara Zannone, Umar Mohammed, and Adriano Koshiyama. Uncovering bias in face generation models. *arXiv preprint arXiv:2302.11562*, 2023.

[230] Vongani H Maluleke, Neerja Thakkar, Tim Brooks, Ethan Weber, Trevor Darrell, Alexei A Efros, Angjoo Kanazawa, and Devin Guillory. Studying bias in gans through the lens of race. In *European Conference on Computer Vision*, pages 344–360. Springer, 2022.

[231] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[232] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM Transactions on Graphics (TOG)*, 41(4):1–13, 2022.

[233] James W Tanaka, Markus Kiefer, and Cindy M Bukach. A holistic account of the own-race effect in face recognition: evidence from a cross-cultural study. *Cognition*, 93(1):B1–B9, 2004.

[234] Christian A Meissner and John C Brigham. Thirty years of investigating the own-race bias in memory for faces: A meta-analytic review. *Psychology, Public Policy, and Law*, 7(1):3, 2001.

[235] Doris Y Tsao and Margaret S Livingstone. Mechanisms of face perception. *Annu. Rev. Neurosci.*, 31:411–437, 2008.

[236] Allison Koenecke, Andrew Nam, Emily Lake, Joe Nudell, Minnie Quartey, Zion Mengesha, Connor Toups, John R Rickford, Dan Jurafsky, and Sharad Goel. Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences*, 117(14):7684–7689, 2020.

[237] Chunxi Liu, Michael Picheny, Leda Sarı, Pooja Chitkara, Alex Xiao, Xiaohui Zhang, Mark Chou, Andres Alvarado, Caner Hazirbas, and Yatharth Saraf. Towards measuring fairness in speech recognition: Casual conversations dataset transcriptions. *arXiv preprint arXiv:2111.09983*, 2021.

[238] Abhilash Mishra and Yash Gorana. Who decides if ai is fair? the labels problem in algorithmic auditing. *arXiv preprint arXiv:2111.08723*, 2021.

[239] Zion Mengesha, Courtney Heldreth, Michal Lahav, Juliana Sublewski, and Elyse Tuennerman. "i don't think these devices are very culturally sensitive."—impact of automated speech recognition errors on african americans. *Frontiers in Artificial Intelligence*, page 169, 2021.

[240] Siyuan Feng, Olya Kudina, Bence Mark Halpern, and Odette Scharenborg. Quantifying bias in automatic speech recognition. *arXiv preprint arXiv:2103.15122*, 2021.

[241] Shikha Bordia and Samuel R Bowman. Identifying and reducing gender bias in word-level language models. *arXiv preprint arXiv:1904.03035*, 2019.

[242] Moin Nadeem, Anna Bethke, and Siva Reddy. Stereoset: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456*, 2020.

[243] Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*, pages 6565–6576. PMLR, 2021.

[244] Olivia Wiles, Isabela Albuquerque, and Sven Gowal. Discovering bugs in vision models using off-the-shelf image generation and captioning. *arXiv preprint arXiv:2208.08831*, 2022.

[245] Joshua Vendrow, Saachi Jain, Logan Engstrom, and Aleksander Madry. Dataset interfaces: Diagnosing model failures using controllable counterfactual generation. *arXiv preprint arXiv:2302.07865*, 2023.