Audio Deepfake Detection Based on Differences in Human and Machine Generated Speech

Submitted in partial fulfillment of the requirements for
the degree of

Doctor of Philosophy
in
Electrical and Computer Engineering

Yang Gao

B.S. Bioengineering, Huazhong University of Science and Technology
M.S. Biomedical Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

May, 2022

To my family

Abstract

Recent advances in deep learning have unfortunately advanced the quality of *Deepfakes* – entirely synthetic multimedia that provide a pernicious means of committing a wide variety of fraudulent activities such as identity theft and spreading misinformation. Deepfake – a portmanteau of "deep learning" and "fake" – is a term that refers to fake media content that is generated or manipulated using deep learning and machine learning algorithms, with the intent to deceive the observer into accepting it as a representation of reality. Amidst growing observation of its misuse and capacity to dilute authentic information, it is of utmost importance that we work towards developing automated systems that can reliably detect deepfakes, so that they can be taken out of circulation before any damage is done.

The term *Deepfake* includes the syndissertation of fake data in all four digital modalities: audio, video, images and text. Deepfakes involving audio, specifically human speech, can be particularly dangerous because of the extensive biometric usage of speech in the world today. Speech systems, especially speaker identification and verification systems, are used to enhance the security of online and telephone-based access control to banking and many other portals. These systems can be attacked using spoofing audios. The related techniques may include audio replay, synthetic speech generation, voice conversion, *etc.* Many of these techniques can be viewed as variants of voice disguise, meant to conceal the speaker's identity by impersonating the target or appearing to be a different person. Such disguise-based techniques are often also encountered in voice-based crimes such as vishing, attempts to break into voice authentication systems, fraudulent calls, *etc.* Voice disguise in itself poses a great threat to automated voice biometric systems, and creates a difficult challenge to forensic speech analysis. Deepfakes deteriorate these problems by enhancing the effectiveness of voice disguise. With the advancement of deep learning techniques, especially the generative models such as generative adversarial networks and WaveNet models, the quality of synthetic speech is steadily becoming closer to a natural speech.

The objective of this dissertation is to develop robust deepfake-speech detection algorithms that can capture the fundamental differences between fake and genuine speech, i.e., between machine-generated and human-generated speech. The algorithms developed must be trainable with limited training data and be adaptable to the latest generation techniques as they are introduced. To achieve this goal, we divide our research into two main tasks, each geared towards answering two fundamental questions as follows:

1. **Section I:** What are the aspects of human speech that deepfake generation techniques *cannot* reproduce?
   - **Part 1: Unique to humans:** What are the characteristics of human speech that are most indicative of the underlying bio-mechanical process of speech production in humans?
   - **Part 2: Unique to machines:** What are the characteristics of machine-generated speech that are most indicative of the underlying algorithmic (computational) process of speech generation?

2. **Section II:** What kinds and categories of models and features are likely to be most adaptable to different deepfake generation mechanisms?
   - **Part 3: Robustness of detectors:** How can we find and use the features that are specific to the human speech production process and least reproducible by machines to build robust deepfake detection techniques?
   - **Part 4: Adaptability of detectors:** How can we develop deepfake detection techniques that can be rapidly adapted to new attacks?

Accordingly, this dissertation has two sections and four parts as mentioned above. The first part (Chapter 2) aims at addressing human-generated speech and tries to identify its unique characteristics from a voice-production perspective. The goal is to understand the human voice production so that later we could identify the most human-centric features that

are not included in the deepfake generative processes. The second part (Chapter 3) discusses machine speech generation mechanisms so that we could later find signatures that are unique to machine-generated speech. For this, we study the mechanisms of both state-of-the-art voice conversion/transformation, and voice synthesis (text-to-speech) systems.

The results of the two sets of studies are finally combined to identify the aspects of machine-generated speech that are simply not consistent with the counterparts expected in human speech. Then, they are used in developing features and models for deepfake detection – a topic addressed by the last parts of this dissertation.

The third and fourth parts are in the Chapter 4 of this dissertation, which deals with developing features and models based on the observation from the first and second parts of this dissertation. Several features have been proposed to improve the robustness of detectors for deepfake detection and their adaptability to newer unseen attacks have also been studied and discussed. In Chapter 5, we summarize the findings and contributions of this dissertation and discuss the future directions of audio deepfake detection.

Acknowledgements

I want to express my ultimate gratitude to my advisors Prof. Rita Singh and Prof. Bhiksha Raj. They always gave me support at the time that I needed it most. Without their support, I could not have had the precious opportunity to spend my youth years on the CMU campus. I enjoyed chatting with Prof. Rita and learned a lot from our discussions. I enjoyed visiting Prof. Bhiksha's office and sharing chocolates and brainstorms. I missed all the weekday meetings, holiday gatherings, office 'parties', and precious paper rewriting time that we spent together. They established a welcome and warm family for all their students, which I sincerely appreciate.

I would also like to thank my doctoral committee members, Dr. Zhizheng Wu, and Dr. Gary Overett. They play important roles and provide me cricial comments and discussions in my dissertation writing and presentation. I am grateful to have them on my committee for this dissertation which builds me up and brings me countless new career opportunities.

Furthermore, I would like to express my gratitude to the professors I met at CMU. I attended their group meetings, appreciated their wise advice, and took precious learnings from them. I would like to thank Prof. Richard Stern, who organized countless group meetings for us. I enjoyed and will always miss the signal processing classes I took from him. I would like to thank Prof. Kumar Bhagavatula, Prof. Aswin Sankaranarayanan, Prof. Marios Savvides for being in my qualify committee and providing me helps and guidances through my preparations. I also want to thank Prof. Prahlad Menon and Prof. Jessica Zhang who enabled me to start this journey by giving me opportunities to start my research (master's) in her lab.

Next, I also want to thank all my fellows and friends who nourished me during my research journey. I want to thank my lab-mates and friends: Yandong Wen, Tyler Vuong, Raymond Xia, Wenbo Zhao, Wenbo Liu, Anurag Kumar, Guan-lin Chao, Benjamin Elizalde, Hira Yasin, Mahmoud Ismail, Shahan Menon, Abelino Jimenez, Maria Joana Correia, Wen Wang, Jinglun Li, Jie Mei, Jun Qi, Yang Zou, Jiyuan Zhang, Chaojing Duan, *etc.* They form the research community that I am most proud of being one inside. I sincerely appreciate my enjoyable time

to meet them, work with them, and learn from them.

I also want to thank the mentors and teammates that I am fortunate to have during my internships at Baidu Research, Facebook and AI Foundation. It was a pleasure to work with Jitong Chen, Kainan Peng, Wei Ping, Qing He, Weiyi Zheng, Zhaojun Yang, Thilo Köhler, Christian Fuegen, Anjali Menon, Buye Xu, Mahsa Elyasi, and Gaurav Bharaj as well as many others. They have great passions in the projects and put time in mentoring me. With their care and inspirings, I was able to grow with their examples.

Lastly, I would like to thank my parents, Cuiling Yang and Feng Gao, for their nearly 'blind' confidences and unconditioned love and supports on me. I have no upper-bound in their eyes. I would also like to thank my husband Jian Wang, for his clear mind, fair advice, and helpful criticisms. I appreciate his accompany, many times as an extra mentor, who did a great job constraining my lower-bound when I fight with my own weakness.

I would like to acknowledge the funding supports from Prof. Rita Singh (the chair of my doctoral committee) and Prof. Bhiksha Raj (member of my doctoral committee), CMU ECE department, the Schmidt Sciences (Palo Alto, CA), and CMU graduate education office.

My Ph.D. journey is about learning of how to fight with unexpected challenges and resolve my continuous self-doubts. All those challenges and self-doubts were painful for me but I learn to finally recognize myself and grow beliefs. I grow love, passion, and determination for work and research. Adversity makes one stronger, just as polishing makes jade finer.

Contents

1 Introduction **1**

1.1 Voice disguise and deepfake speech 2

1.2 Threats from deepfake speech 4

1.2.1 Spoofing threats to automatic speaker verification systems . . 4

1.2.2 Comparing human- and machine-generated speech in spoofing attacks 6

1.3 Objectives of the thesis 13

2 Human speech **15**

2.1 The human voice production mechanism 15

2.2 Characteristics of human-generated speech 17

2.2.1 Non-speech signatures in natural human voice 17

2.2.2 Breath sounds and their potential use 17

3 Machine-generated speech: methods of generating deepfakes **21**

3.1 Voice transformation 23

3.1.1 Generative Adversarial Networks (GANs) 26

3.1.2 GANs for voice mimicry 30

3.2 Speech synthesis 36

3.2.1 Prosody transfer 37

3.2.2 Voice synthesis with style characteristics 48

3.3 Key insights 61

4 Deepfake detection for practical systems **65**

4.1 Features that capture signatures of human-generated speech 65

4.1.1 Human voice-production-based features 66

4.1.2 Analyzing the robustness of voice-production-based features . 68

4.1.3 Key insights 71

4.2 Features that capture signatures of machine-generated speech 72

4.2.1 Artifacts in machine-generated speech 72

4.2.2 Generalized spoofing detection inspired from audio generation artifacts 74

4.3 Data-driven approach for deepfake detection 87

4.3.1 Advanced deep learning models and the issue of generalization 87

4.3.2 Inductive learning for deepfake detection 88

4.3.3 Data augmentation for deepfake detection 91

5 Conclusions **93**

5.1 Summary and contributions 93

5.2 Future directions and remaining challenges 96

5.2.1 Data collection 96

5.2.2 Filled, unfilled pause and breath 96

5.2.3 Segmental-level audio deepfake detection 97

List of Figures

1-1 A depiction of the relationships between voice disguise, audio deepfake and spoofed audio 3
1-2 Feature space for D-vectors of speaker verification 11
1-3 Objectives 14

2-1 Parts of the vocal tract [1] 15

3-1 The original GAN model 28
3-2 Style transfer by GAN 28
3-3 The DiscoGAN model 29
3-4 Visualization of Generator G_A and Discriminator D_A 32
3-5 The proposed VoiceGAN Model 33
3-6 Visualization of the spectrograms through the voice transformation . 34
3-7 A simplified flowchart of Baidu DeepVoice3 system 40
3-8 Training phase of style tokens [2] 42
3-9 Our proposed reference encoder model (left zoomed in) 43
3-10 The same utterances inferenced using four different tokens learned through the TTS training with VCTK dataset. 45
3-11 The same utterances inferenced using token No.1-5 46
3-12 F0 visualization of synthesized utterance using different tokens 47
3-13 The same utterances syntheized using different prosodies 48
3-14 Visualization of F0 of two different utterances 49
3-15 Multimodal style extraction model 54
3-16 Style embedded TTS framework 55

4-1 Spectral envelopes of real speech and fake speech 68
4-2 Spectral entropy distributions. Blue is for fake speech and organge is for real speech 70
4-3 Artifacts in the text-to-speech [3] 73
4-4 Artifacts in the voice conversion [3] 73
4-5 Checkboard effects in feature generation 76
4-6 Block diagram of the baseline system (left) and the zoomed-in view of one residual block (right) 80
4-7 Visualization of the proposed features averaged within different spoofing types 81

List of Tables

1.1 EERs of impersonation attacks to the ASV under black-box scenario 9

1.2 EERs of evaluation set for ASVspoof 2019 LA under black-box and white-box scenarios 12

3.1 NIST STNR test 35

3.2 Training data style label statistics 54

3.3 Style classification on TTS data 57

3.4 TTS data F0 statistics 58

3.5 Feature selection 59

3.6 Subjective preferences 60

4.1 ASV countermeasure-based evaluation 69

4.2 SpecAugment (SA) and normalization approaches 82

4.3 Single system comparisons as ASV countermeasures 82

4.4 Weighted voting scores with different voting mechanisms 84

4.5 EERs of evaluation set for ASVspoof 2019 LA for speaker verification 84

4.6 Breakdown analysis of the performance on different spoofing audio types 85

4.7 EERs of evaluation set for ASVspoof 2021 DF track 90

4.8 Performance of the inductive model for ASVspoof 2021 DF track . . . 90

Chapter 1

Introduction

The term “Deepfake” is a portmanteau of the words “deep learning” and “fake”. Deepfakes represent fake audio,video,image or textual media content that is generated or manipulated using deep learning and machine learning algorithms, with the intent to deceive. With the recent advances in deep learning, the quality of deepfakes has significantly improved, making them an even more pernicious means of committing a wide variety of fraudulent activities, such as identity theft and spreading misinformation. For example, in 2019 a UK company’s CEO was scammed over the phone to transfer 220,000€ into a bank account. This fraudulent call used deepfake audio that was generated to successfully impersonate the voice of the parent company’s CEO. There have been multiple such cases that impress upon us the gravity of the threat that deepfakes pose to society. Furthermore, deepfakes are identified as a particularly potent threat their generation requires minimal manual effort. Thus an attacker with sufficient computational resources can generate fake data on a massive scale relatively easily. With the threat from deepfakes, the general public’s trust in media content is being increasingly diminished and destroyed. Therefore, it is of utmost importance that we work towards developing automated systems that can reliably detect deepfakes, so that they can be taken out of circulation before any damage is done.

Deepfakes involving speech, as in the example above, can be particularly dangerous because of the extensive biometric usage of speech. Even before deepfakes came to the fore, attackers had been using a wide variety of techniques to spoof speech systems. These techniques included various forms of voice disguise and emulation of statistical properties of speech [4, 5]. Synthesized speech was being used to fool voice authentication systems, and forged audio recordings were being used to defame public figures [6, 7].

1.1 Voice disguise and deepfake speech

Speech systems can be spoofed in many ways, depending on how they work. The techniques used may include audio replay, synthetic speech generation, voice conversion, *etc.* Many of them can be viewed as voice disguise meant to conceal the speaker's identity by impersonating another person or appearing to be a different person. Such disguise-based techniques are often encountered in voice-based crimes such as vishing, attempts to break into voice authentication systems, fraudulent calls *etc.* Voice disguise poses a difficult challenge to forensic speech analysis.

Voice disguise is defined in [8] as "any alteration, distortion and deviation from the normal voice, irrespective of the cause". In other words voice disguise can be construed as the deliberate distortion of voice by the speaker to change its perceived identity. Voice disguise can be categorized along two dimensions: intention and method. Intention may be either "deliberate" or "non-deliberate". The method used may be "electronic" or "non-electronic" based on whether or not electronic (or digital) techniques have been used to alter ther voice. Electronic methods are often intentional, and may involve the use of scrambling devices to alter the voice; however they may also be non-deliberate and distortions and alterations may be introduced by the limitations of the equipment used to produce, record or transmit voice, *e.g.*,

bandwidth limitations of telephones, or linear/non-linear or frequency characteristics of recording equipment.

The influence of voice disguise on the performance automatic speech recognition and speaker verification systems has been widely studied [5, 9]. In [5], voice disguise and automatic speech recognition (ASR) are studied and discussed. A disguise voice dataset was created by recording 10 types of deliberately non-electronically disguised voice speech of twenty male student speakers. The 9 disguised-voice methods have great effect on the performance of the speaker verification system while the foreign accent affects the ASR the least and whispering and masking of the mouth affects ASR the most. [9] discussed the voice disguise in ASR thoroughly as a review paper. Apart from some conventional electronic voice disguise such as electronic scrambling devices, voice conversion and text-to-speech synthesis have been specifically discussed as a recently common way in deliberate-electronic voice disguise. In the context of this thesis we focus on disguise through digital electronic means, such as conversion or wholesale synthesis, both of which can be achieved through *deepfake* systems.

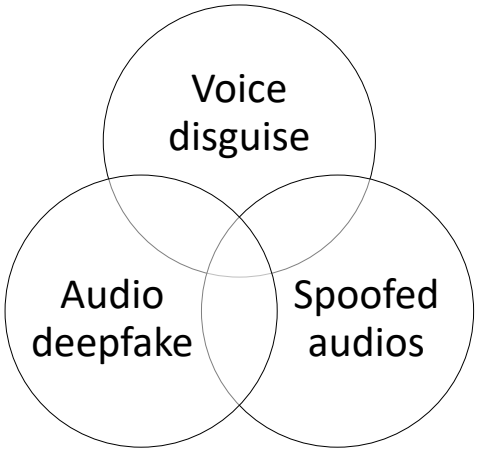


Figure 1-1: A depiction of the relationships between voice disguise, audio deepfake and spoofed audio. In recent times, audio deepfakes are increasingly among the most damaging and costly techniques being applied in the voice disguise and spoofed audio domains.

1.2 Threats from deepfake speech

With the advancement of deep learning techniques, especially generative models such as generative adversarial networks and wavenet models [10], the quality of synthetic speech is getting much closer to natural speech [11]. The threats from such skilled synthesis arise from all speech technologies, including speech recognition and downstream speech-based services, but, in today's technological setting, perhaps the greatest threat is to speech biometric systems, in particular automatic speaker verification (ASV) systems. This is the focus of our work.

1.2.1 Spoofing threats to automatic speaker verification systems

Automatic speaker verification (ASV) systems utilize the biometric information in human speech to verify the identity of a speaker by matching it with the information present in a database (which is also derived from speech samples). As an increasingly popular and common biometric authentication mechanism used as "gateways" to various services, they are particularly attractive as a target for "spoofing" attacks that attempt to induce the ASV system to return wrong results, allowing an impostor to bypass the system and gain access.

There are many spoofing methods in use by attackers nowadays, including direct human impersonation of the target, audio replay, machine assisted-speech generation such as voice conversion (VC), customized and manipulated text-to-speech synthesis (TTS) system outputs, *etc.* Among these spoofing methods, perhaps the greatest threat comes from synthetic speech produce by modern deep learning systems. With the advancement of deep learning techniques, especially with advancements in generative models such as generative adversarial networks and wavenet models [10], the quality of synthetic speech is getting much closer to natural speech [11]. The tech-

nology too is easily available to anyone, with open-source tools available on the web, making this possibly the fastest-growing and most dangerous threat to ASV systems.

The ASVspoof challenge series [12] have raised efforts in fake speech spoofing attack countermeasures on ASV systems. The key challenge here is to explicitly or implicitly detect if a speech sample (provided for ASV) is spoofed. Previous studies on anti-spoofing attacks on ASV systems and synthetic speech detection have evaluated different features [13] and deep learning models [14] for detection performance. However, with the fast evolution of deepfake techniques, developing a detection system that is not constrained by the training data and can accurately detect new spoofed data generated from different or unseen deepfake algorithms is still a challenge.

This addresses the challenge of distinguishing between fake/synthesized and human-generated speech. We start from the hypothesis that human- and machine-generated speech have many differences in many aspects. It is important to establish these differences and use them for the robust detection of deepfake speech.

As a first step, in the next section, we compare the threats from impersonation attacks with synthetic speech attacks and establish that deep synthesized fakes are in fact the most dangerous attacks for ASV systems as published in [15].

Prior work

A number of approaches of varying success have been proposed in the literature to detect fake speech to increase the security of ASV systems against spoofing attacks. For example, the long-running ASVspoof challenge [16, 17, 12] has raised wide efforts in fake speech spoofing attack countermeasures on ASV systems. The main focus of the challenge, however, has been to rank spoof detection countermeasures, and not to carry out an in-depth evaluation of the ASV systems' performance under attacks. Another significant problem with the spoof detection study is that, with the rapid evolution of deepfake generation methodologies, the sheer variety of attacks that an

AVS system may be subject to is also rapidly increasing. Detection models trained on a specific provided dataset synthesized using a limited set of methods are unlikely to generalize to newer types of fake/generated audio. In ASVspoof2019, for example, detection algorithms [14, 18, 19, 20] that work very well on training datasets are often found to perform much worse on evaluation sets that have been produced using attack techniques not present in the training data. The detection performance on the evaluation sets could be an indicator of the generalization capacity of those proposed algorithms as one purpose of the challenge.

To include the latest deep-learning speech synthesizers, [21] provides a synthetic speech dataset called Fake or Real (FoR), improving variety of the deepfake speech data for this purpose. We have, in fact, also used it effectively in the context of this work.

1.2.2 Comparing human- and machine-generated speech in spoofing attacks

In this section, we compare the threats from impersonation attacks with synthetic speech and establish that deep synthesized fakes are in fact the most dangerous attacks for ASV systems.

Datasets

For our experiments we use four datasets: the logical access (LA) of ASVspoof 2019 dataset [12], the VoxCeleb dataset [22], the FoR dataset [21] and our own collected impersonation dataset (CID).

The CID dataset is collected from the performances of expert impersonators on YouTube, and segmented carefully to only keep the speech segments corresponding to target speakers. All impersonators in the CID dataset are professionals mimicking political figures for amusement, collected from TV shows and talk shows on YouTube.

The dataset comprises 1091 utterances of genuine speech from both the impersonators and the political figures and 981 utterances of impersonated speech. The data are further segregated into paired sets, each pair containing an impersonator's real speech and target/mimicked speech, or the target's real speech and the speech produced for the same target by an impersonator. These are indicated in Table 1.1, in which pairs that originate from the same speaker are called positive pairs, and pairs from different speakers are called negative pairs.

As shown in Table 1.1, we have two sets of positive pairs and four sets of negative pairs. There are 19086 positive pairs of same target speaker's real utterances (R), 14844 negative pairs of different target speaker's real utterances (RI), 3382 positive pairs of same impersonator's impersonations for different target (IAB), 37554 negative pairs of target and impersonation pair (TI), 1988 negative pairs of different impersonator's real utterances (IRAB) and 28080 negative pairs of target/impersonator's real utterance pair (IRT).

For synthetic data, ASVspoof2019 dataset contains logical access data and physical access data. In this study, we only use the logical access data which contains machine-generated speech using multiple text-to-speech synthesis and voice conversion methods. The logical data has 2580 bonafide utterances and 22800 synthetic utterances from 20 speakers in the training set; 2548 bonafide utterances from 20 speakers and 22296 spoof utterances from 10 speakers in the development set [12]. The evaluation set contains 7355 bonafide utterances from 67 speakers and 63882 spoof utterances from 48 speakers. The spoof audio are generated using unseen spoofing algorithms intentionally, aiming to give insights of the generalization performance of the proposed countermeasure models. In order for a general ASV system to evaluate this dataset, we generate 4914 bonafide positive pairs and 4914 negative pairs for each attack (A07-A19) from the original evaluation set. We also generate 15970 positive pairs and 15970 negative pairs to evaluate the overall attacking ability

over all attacks.

To best evaluate the threats of different attacks, we train ASV systems under unconstrained recording and speaking conditions (essentially data-in-the-wild). For this, we use the VoxCeleb dataset, which is a large-scale public dataset containing millions of utterances collected from unconstrained speech samples [22]. It has many speakers and millions of utterances under different recording conditions. This can be effectively used to evaluate the potential of any given ASV methodology to generalize to unseen speakers and unconstrained conditions [22, 23].

Analyzing performance under attacks on black-box and white-box ASV systems

The ASV model we use is proposed by Chung *et al.* (2020) [23], which applies the Thin ResNet-34 [22] as backbone, and Self-attentive Pooling(SAP)[24] as aggregation strategy. This model, when trained with short-time Fourier transform (STFT) spectrograms of Voxceleb recordings generalizes extremely well to unconstrained conditions as shown by the low EER of real utterance pairs, mentioned earlier in this section.

The **black-box** ASV system is pretrained with the VoxCeleb dataset. STFT, MFCC, aperiodic parameters (AP) and spectral envelope (SP) are used as input features to this model. The original input audio comprise segments of two seconds duration. We use the same STFT feature as in [23, 22]. MFCC features are computed from 16kHz sampled signals. They comprise 13 cepstral coefficients, to which first and second-order derivatives respectively are concatenated, making the feature dimensionality 39. (AP and SP are not the focus of this section and will be further discussed in Section.4.1.1 and Section.4.1.3)

The **white-box** model is trained with the ASVspoof 2019 data, as a multi-class classifier for speaker identification. We make small modifications to the initial

Table 1.1: EERs of impersonation attacks to the ASV under black-box scenario

		ASV EER%							
Impersonation Data		R[1]+RI[2]	IAB[3]+RI	R+TI[4]	IAB+TI	R+IRAB[5]	IAB+IRAB	R+IRT[6]	IAB+IRT
blackbox	VoxCeleb2(STFT)	1.71	13.30	11.42	43.52	4.86	17.76	5.21	19.45
	VoxCeleb1(STFT)	4.16	16.41	14.95	42.02	4.74	15.90	5.06	15.64
	VoxCeleb1(MFCC)	17.21	22.09	22.01	48.77	9.22	26.80	8.86	20.11
	VoxCeleb1(AP)	39.75	41.27	44.89	45.46	45.58	46.06	41.65	42.94
	VoxCeleb1(SP)	53.58	49.42	54.36	50.36	55.04	51.11	53.76	49.43

[1] R: Same target speaker's real utterance pair (+, #19086) [2] RI: Different target speaker's real utterance pair (−, #14844)
[3] IAB: Same impersonator, impersonations for different target pair (+, #3382)
[4] TI: Target and impersonation pair (−, #37554) [5] IRAB: Different impersonator's real utterance pair (−, #1988)
[6] IRT: Target and impersonator's real utterance pair (−, #28080)
[7] The model pre-trained with VoxCeleb2 dev set using Spectrogram feature

ASVspoof2019 training set by assigning each spoofed utterance an identity which uniquely incorporates both speaker and attack. There are 20 speakers and 6 types of attack in the ASVspoof2019 LA training set, meaning that there are 120 "spoofed identities". Thus our modified training set contains 140 identities. We call these *ASVspoof training identities* (ASVTIs).

Impersonation attacks

Our black-box evaluations on impersonation attacks use the CID dataset. We run several experiments to evaluate the dataset's attacking potential. The results are shown in Table 1.1. The model that is pretrained on VoxCeleb2 is able to verify open-set speakers best and gives 1.71% EER for target speakers' real utterances (positive and negative pairs R + RI); this pretrained model can be seen as a black box ASV under open-set evaluation.

From our tests, we observe that combining the impersonation/target pairs (TI) with the positive pairs from real speakers (R) improves the speaker verification EER to 11.42%, which indicates that professional impersonation can fool the ASV system to a certain extent, although it is still ineffective in most attacks. The group with real speaker positive pairs (R) and negative pairs (IRAB) built from the real voices of

different impersonators has a low EER of 4.86%, showing that the pre-trained ASV system is indeed generalized to verify unseen target speakers and cross-impersonator pairs.

The IAB is the same impersonator mimicking different targets. IAB + RI has an EER of 13.3%, showing that even if the same speaker tries to impersonate different targets, their utterances are mostly considered as the same speaker, although still having some capacity to fool the ASV system. Note that this EER value gets significantly larger to 22.09% with less generalized models, such as the VoxCeleb1 pre-trained model. This indicates that impersonation from professional impersonators is still threatening to some ASV systems. The IAB + IRAB set has comparable EERs as the IAB + RI, showing that the IRAB pairs are valid, also indicating the true differences between the impersonator's real voices. Moreover, the IAB + TI gives an EER of 43.52%. This high EER comes from the formation of this evaluation set. Different from other sets, both the negative pairs and positive pairs can be seen as the spoofing attacks because the professional impersonator could impersonate different targets to a certain extent, which makes the "positive" pairs negative in nature. Therefore, both the positive pairs of IAB and the negative pairs of TI are the hardest cases, also shown by the EERs of their combination with the R and RI.

The R + IRT corresponds to positive pairs for the real voice utterances of the same targets and negative pairs of impersonator's real voice with the targets' real voice. The 5.21% EER shows that the impersonator's real voices are indeed not similar to the targets' voices.

The overall results show that while mimicry from amateur impersonators is reported to not succeed in fooling ASV systems in previous research [25, 26], mimicries rendered by professional impersonators still poses threats to a certain extent.

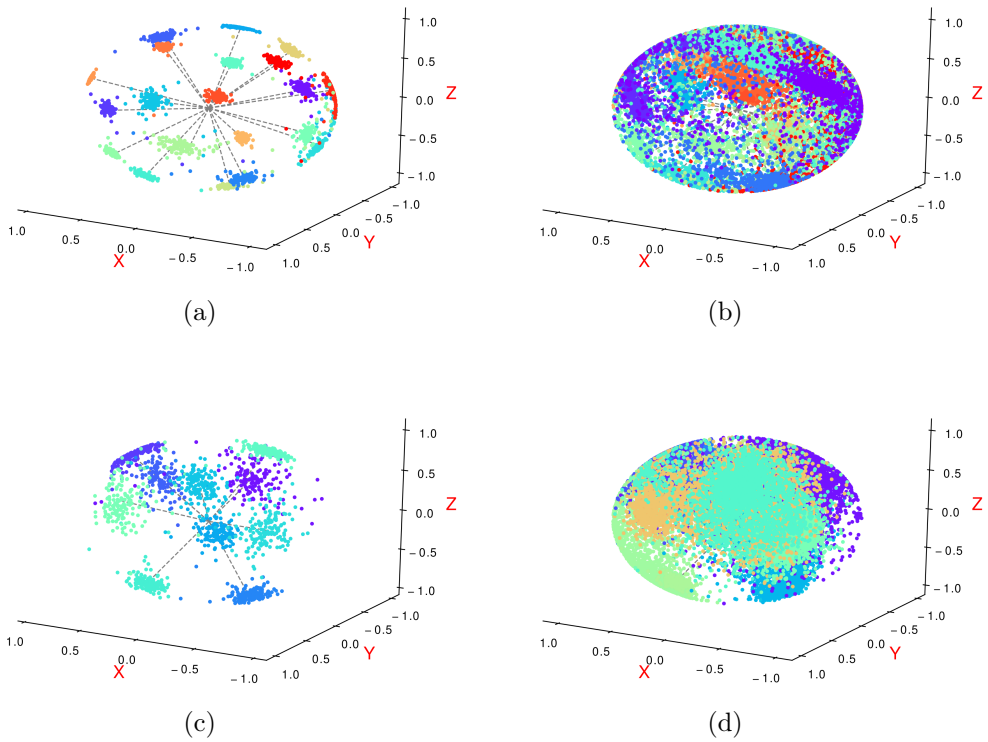


Figure 1-2: Feature space for D-vectors of speaker verification. (a) 20 bonafide speakers in training set; (b) 20 bonafide and spoof speakers in training set; (c) 10 bonafide speakers in development set; (d) 10 bonafide and spoof speakers in development set. The embedding features from the bottleneck layer of the white-box ASV model are mapped on a sphere. Spoofed utterances introduce ambiguity to the feature space.

Table 1.2: EERs of evaluation set for ASVspoof 2019 LA under black-box and white-box scenarios

		ASV EER%													
	Attack	A07	A08	A09	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	ALL[3]
blackbox	VoxCeleb2(STFT)	34.03	23.20	5.70	48.51	37.37	43.42	23.67	40.45	43.14	50.51	4.99	7.10	11.26	21.42
	VoxCeleb1(STFT)	27.93	25.30	11.01	47.77	37.36	44.77	30.93	43.33	40.91	43.36	7.65	10.83	13.97	22.03
	VoxCeleb1(MFCC)	45.12	28.89	16.02	45.01	48.88	45.09	38.13	35.06	43.01	46.04	11.07	25.64	25.02	25.66
	VoxCeleb1(AP)	39.89	24.92	31.63	38.66	21.93	43.05	30.99	28.25	42.21	45.55	31.56	31.92	39.27	35.55
	VoxCeleb1(SP)	51.47	50.73	53.51	51.46	53.97	49.52	54.51	49.92	51.39	50.66	52.71	49.18	54.97	50.60
	Todisco *et al.* (2019), [12]	59.68	40.39	8.38	57.73	59.64	46.18	46.78	64.01	58.85	64.52	3.92	7.35	14.58	-
whitebox	ASVSpoof(STFT)[1]	2.33	2.65	3.75	47.56	40.89	47.59	37.01	29.09	35.48	4.09	12.07	28.61	1.88	22.24
	ASVSpoof(MFCC)	7.12	5.08	8.12	39.76	28.99	49.01	33.81	19.04	41.39	9.08	18.00	16.47	2.09	15.99
	ASVSpoof(AP)	38.93	32.46	32.59	42.37	38.29	43.28	37.02	33.96	41.12	49.06	40.05	34.57	44.53	39.25
	ASVSpoof(SP)	50.97	49.94	40.07	49.75	49.25	52.04	52.30	51.03	51.74	51.99	41.49	46.16	45.78	42.08
	VoxCeleb2+ASVSpoof(STFT)	**1.16**	**2.31**	**0.77**	**43.42**	**27.62**	**41.23**	**15.46**	**34.48**	**36.26**	**6.63**	**1.26**	**5.75**	**0.68**	**11.99**
	VoxCeleb1+ASVSpoof(STFT)	1.21	2.63	1.75	45.85	17.40	45.69	20.84	25.85	25.41	4.66	2.24	8.24	0.73	13.35
	VoxCeleb1+ASVSpoof(MFCC)	4.99	4.51	1.99	37.28	19.02	45.08	33.18	15.92	33.65	6.01	11.30	11.44	2.98	14.94
	VoxCeleb1+ASVSpoof(AP)	22.04	17.77	28.13	33.68	37.78	37.20	19.72	**6.50**	33.16	44.43	33.25	32.33	41.01	32.45
	VoxCeleb1+ASVSpoof(SP)	50.24	44.30	40.51	49.21	48.82	50.45	48.74	48.62	49.43	50.86	33.63	42.10	38.06	36.79

[1] The model trained directly with ASVTIs[4] using Spectrogram feature [2] The model pre-trained with VoxCeleb1 dev set using Spectrogram(blackbox) and subsequently trained with ASVTIs [3] Evaluation on general pairs as described in 1.2.2, indicating overall EER [4] As defined in 1.2.2

Synthetic speech attacks

To further understand the attacks of synthetic speech generated from different methods, we perform extensive ASV evaluations on the ASVspoof evaluation set under the black-box and white-box conditions. The evaluation set contains attack methods from A07 to A19 which are different voice conversion or speech synthesis techniques [27].

Comparing human-generated attacks and machine-generated attacks, as in the black-box scenarios for both cases, we found that the general attack strength of the machine-generated speech is stronger than the human-generated attacks, as shown in Table.1.1 and Table.4.5. For the blackbox of VoxCeleb2 trained using STFT, most of the synthetic attacks have an EER of over 20%, much higher than the EER of impersonation attacks (IAB).

As shown in Table 4.5, A09/A17/A18/A19 are relatively weaker attacks showing lower ASV EER% than STFT/MFCC-based black-boxes, which is consistent with the results given by [12]. These attacks are generated through waveform generators such as waveform filtering and spectral filtering, which may be simpler methods compared to hard cases using neural vocoders. Most of attacks tend to be more dangerous for

MFCC-based black-boxes than STFT-based black-boxes, as STFT usually captures more nuances information compared to MFCC. Also for STFT and MFCC, EERs for most attacks are lower under the white-box scenario, compared to attacks on black-boxes with the same dataset and feature settings. However, it does not result in much improvement of EER for A10, A12, and A15, which are generated by **neural waveform models**, indicating increased threat from deepfakes.

In conclusion, the feature robustness ranks as 'STFT > MFCC' with finetuning under white-box scenario. This conclusion is consistent with our hypothesis that features that capture information about prosodic nuances are more robust under attacks for ASV systems. In Figure 1-2, we draw the embedding features from the bottleneck layer of the white-box ASV model on a sphere. When spoofed utterances are introduced, it is not easy to discriminate the embeddings anymore, which indicates their threats to the ASV system.

Conclusions

The threats shown above from the deepfakes indicates the urgent needs for fake speech detection. As shown in the above study, we have established that spoofing attacks carried out using deep-fake speech are more likely to be effective than those using other synthetic methods or human impersonation, even if the speech is produced by professional impersonators. We need to have more robust detection of spoofed speech, which could result in rendering ASV systems more robust to attacks generated using unseen methods.

1.3 Objectives of the thesis

We are going to develop robust deepfake-speech detection algorithms that can capture the fundamental differences between fake and genuine speech, *i.e.*, between machine-

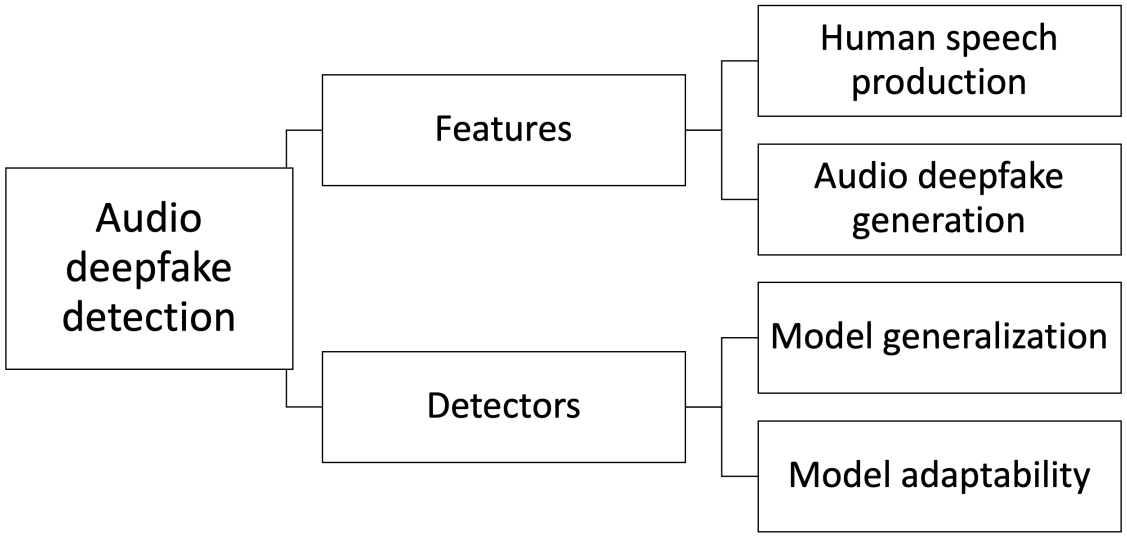


Figure 1-3: Audio deepfake detection based on differences in human- and machine-generated speech

generated and human-generated speech. We aim to develop features that could capture these differences. The algorithms developed must be trainable with limited training data and be adaptable to the latest generation techniques as they are introduced. To achieve so, we will start from looking at the human speech production and the deepfake generation techniques as in the following chapter.

Chapter 2

Human speech

In this chapter, we briefly discuss the bio-mechanical process of speech production in humans, and their relevance to deep-fake detection.

2.1 The human voice production mechanism

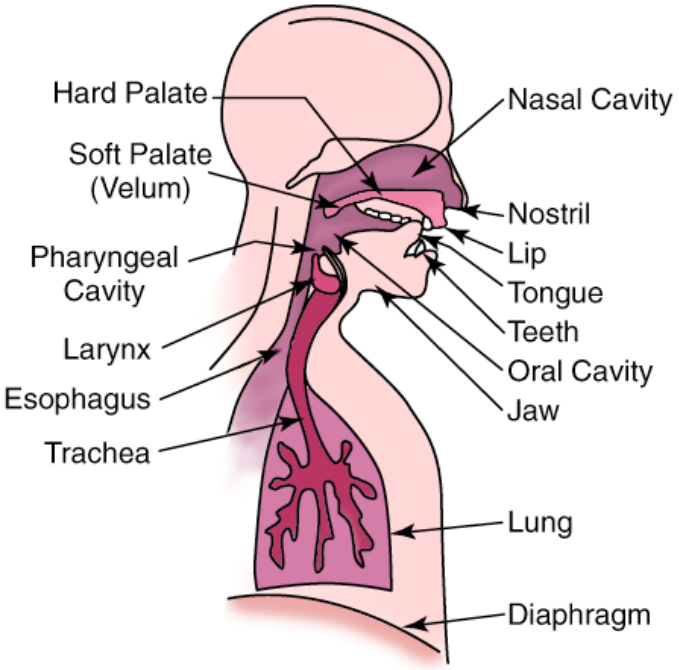


Figure 2-1: Parts of the vocal tract [1]

Voice production is the result of a complex interaction of excitation, articulation and resonance. The vocal tract begins from the vocal cords and ends at the lips, and includes the pharynx and the oral cavity. Most speech production occurs during exhalation: the lung pumps out air, which pushes past the vocal cords and excites the vocal tract resonators such as the throat, mouth cavity and nasal passages. In voiced speech the vocal cords vibrate, resulting in periodic pulses of air, giving voiced sounds their characteristic fundamental tones. These are amplified and modified by the vocal tract resonators to create a person's recognizable voice. For unvoiced sounds the vocal cords remain open, and the exhaled air creates turbulences that excite the resonators. In all cases, the tongue, soft palate and lips further modify the sounds to produce recognizable words in the person's characteristic voice.

In more detail, the speech production mechanism for voiced sounds such as vowels in particular works as follows:

1. Air enters the lungs via normal breathing.
2. As air is expelled from the lungs via the trachea, the tensed vocal cords within the larynx are made to vibrate by the variations of air pressure in the glottal opening.
3. The glottal orifice opens and closes, modulating the air flow into quasi-periodic pulses.
4. These pulses are frequency-shaped by the throat/mouth/nasal cavity. The positions of the various articulators (jaw, tongue, velum, lips, and mouth) determine the produced sound.

2.2 Characteristics of human-generated speech

2.2.1 Non-speech signatures in natural human voice

In human-generated speech, the signals carry information about the speaker. Both voiced and unvoiced speech have already been widely used to analyze the information carried in speech [28]. Meaningful features can be extracted from it, such as pitch, formants, jitter, shimmer, energy, loudness, zero-crossing rate, spectral entropy, *etc.* Furthermore, with large prosody/emotion-labeled datasets and state-of-the-art machine learning models, "deep" features associated with less explicit attributes of the speech can also be extracted through well-architected machine learning models [29, 30].

However, there is one aspect of voice that has not garnered sufficient attention – the non-speech parts of the signal. This includes the breath sounds, background noise, filled pauses *etc.* While these are sometimes considered in utterance-level machine-speech generation [31, 32], particularly filled pauses in spontaneous speech which are synthesized with rigorous patterns [33, 34, 35], their utility as biometric identifiers is generally ignored.

2.2.2 Breath sounds and their potential use

As discussed above for the speech production mechanism, speech is largely produced during exhalation. In order to replenish air in the lungs, speakers must periodically inhale. When inhalation occurs in the midst of continuous speech, it is generally through the mouth. Intra-speech breathing behavior has been the subject of much study, including the patterns, cadence, and variations in energy levels. However, an often ignored characteristic is the *sound* produced during the inhalation phase of this cycle. Intra-speech inhalation is rapid and energetic, performed with open mouth and glottis, effectively exposing the entire vocal tract to enable maximum intake of air.

This results in vocal tract resonances evoked by turbulence that are characteristic of the speaker's speech-producing apparatus. Consequently, the sounds of inhalation are expected to carry information about the speaker's identity. Moreover, unlike other spoken sounds which are subject to active control, inhalation sounds are generally more natural and less affected by voluntary influences.

In [36], we demonstrated that breath sounds are indeed bio-signatures that can be used to identify speakers. In this paper, we used Sphinx-3, a state-of-art Hidden Markov Model (HMM) based automatic speech recognition (ASR) system to first obtain accurate phoneme segmentations for all the speech included in the LDC Hub-4 1997 Broadcast news database [37]. The database comprises single-channel recordings of read speech from multiple news anchors and people interviewed within the news episodes. The recordings are sampled at 16000 Hz. The ASR system was trained on this database, and the acoustic models obtained were used to obtain highly accurate phoneme segmentations. Breath was modeled as a phoneme during the training process, and thus the process of phoneme segmentation directly yielded the breath sounds that we needed for our experiments.

The complete set of breath sounds extracted from the Hub-4 database included more than 3000 combinations of speaker, channel (broadband and telephone), fidelity (high, low, medium) and type of speech (read and conversational), of which we only chose breath sounds that corresponded to high fidelity clean read speech signals for our experiments. Since the goal of this paper is confined to demonstrating that breath can indeed be used to identify speakers, we did not attempt to explore of performance in different speech styles, channel types, noise conditions, *etc.*

We used two feature formulations, i-vectors, and a set of novel CNN-RNN based features derived from constant-Q representations of the speech signal. Experiments with both i-vectors and constant-Q spectrograms show that breath sounds can be successfully used to identify speakers. In fact, for the clean speech signals that we

used in our experiments, the accuracies are surprisingly good. Note that since our primary objective in this paper is to demonstrate that the sound of the human breath can indeed be used for speaker identification, this choice of features was judiciously made to fulfill our goal of providing proof-of-concept.

The CNN-LSTM neural network based framework proposed in our paper uses constant-Q representations to effectively normalize the shifts in the resonance patterns of breath within the same speaker and to emphasize the distinction between speakers. The CNN-LSTM based approach automatically learns shift-invariant and temporal features, and combines feature extraction, speaker modeling and decision making into a single pipeline. This framework is also distribution-assumption free and works effectively for short recordings. Results show that it works better than the i-vector based classifier and achieves high accuracy in the speaker identification through breath sounds. At the same time, both features have the area under the curve (AUC) value of more than 0.94 of their respective receiver operating curves (ROC). More experimental details could be found in [36].

From the experiments, we showed that these sounds by themselves can yield remarkably accurate speaker recognition with appropriate feature representations and classification frameworks. This is interesting since breath is a factor that has not been well-modeled in deepfake speech generation [32] and could be leveraged for detection purposes.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 3

Machine-generated speech: methods of generating deepfakes

In this chapter, we will discuss generation methods for deepfake speech and insights to improve the detection of audio deepfake. The goal is to understand the generation process so that we could gain insights about the characteristics of machine-generated speech that are most indicative of the underlying algorithmic process of speech generation.

Deepfake speech generation methods have become increasingly sophisticated with advances in techniques in big data, machine learning, deep learning, and graphic processing units (GPUs) in recent years. In principle, there are two representative types of deepfake-speech generation methods: *voice conversion* and *speech synthesis*. In voice conversion (or voice transformation) a speech recording from one speaker is qualitatively converted to sound like another while preserving linguistic information. On the other hand, speech synthesis is the artificial production of human speech by rendering text or symbolic linguistic representations like phonetic transcriptions into speech. Both approaches are widely used to generate deepfake audio that could deceive listeners.

To create near-authentic deepfake audio, the voice characteristics and speaking style of the target speaker need to be considered in the generation. For example, the emotion, pitch, speaking rate, pause, emphasis, *etc.*, are taken into accounts in many related studies [38].

The progress of deep learning techniques has resulted in advances in both voice conversion techniques and speech synthesis. Understanding these techniques is an essential step to develop well-adapted detection methods of deepfake speech. In order to understand how voice conversion could achieve quality that could be consdered a deep-fake threat, we studied the voice *impersonation*, which goes beyond mere voice *conversion*. Unlike voice conversion, which only captures immediate frequency characteristics of the target speaker, impersonation also attempts to mimic other aspects such as pronunciation and prosody in order to achieve a higher level of deception. In the *text-to-speech synthesis*, the most advanced works focus on pushing the boundary of synthesized speech quality both in terms of the voice characteristics of the target speaker, and their prosody and style. Prosody, here, refers to the elements of speech beyond the individual phonetic segments, encompassing properties of syllables and larger units of speech, such as linguistic elements, and intonation including tone, stress and rhythm.

In the field of deep generated/changed speech, there is surging research on topics such as controllable TTS, one-shot voice conversion, and so on [39]. However, based on investigations into approaches that are very commonly used in deepfake audio generation, we found several places that could be potential weakness that could be exploited for the robust detection of deepfakes.

Breath sounds are hardly considered in speech conversion or synthesis since currently, related research mainly focuses on production of short sentences (that are too short to require intra-sentence inhalations or exhalations), rather than long or extended passages. Besides breath sounds, other subtle transitions between utterances

are also not considered, for instance the termination of a sentence is often represented by complete silence or random noise that may not be consistent with the speech or background noise. Furthermore, depending on the algorithms and models that are used to generate or convert speech, there may be some deep-fake signature artifacts or distortions in feature space that could be detectable by machines but are not distinguishable for human listeners. In understanding the generation process of deepfake speech, we can use the insights to develop better models for the detection of deepfake speech, as in Section 4.2.

3.1 Voice transformation

Deepfake audio aims at deceptive fake audio that has high quality and near-authentic imitation of real audio. Voice transformation aims to transform the source speaker's audio to the target speaker's voice while keeping all the linguistic information. With deeplearning techniques getting increasingly powerful, deepfake audio from voice transformation using deep learning techniques have become one of the primary means of creating deceptive fake audio recordings.

Traditionally, for voice transformation, individual frames of the source speaker's speech are warped to match the pitch and other tones of the target's speaker's voice. However, this creates unnatural effects and inconsistent prosody in the conversion results. Deepfake audio, on the other hand, results in high quality impersonation of the target's speaker.

Deepfake voice impersonation is not the same as voice transformation, although the latter is an essential element of it. In voice impersonation, the resultant voice must convincingly convey the impression of having been naturally produced by the target speaker, mimicking not only the pitch and other perceivable signal qualities, but also the style of the target speaker. In this chapter, we propose a novel neural-network

based speech quality- and style-mimicry framework for the synthesis of impersonated voices. The framework is built upon a fast and accurate generative adversarial network model. Given spectrographic representations of source and target speakers' voices, the model learns to mimic the target speaker's voice quality and style, regardless of the linguistic content of either's voice, generating a synthetic spectrogram from which the time-domain signal is reconstructed using the Griffin-Lim method. In effect, this model reframes the well-known problem of style-transfer for images as the problem of style-transfer for speech signals, while intrinsically addressing the problem of durational variability of speech sounds. Experiments demonstrate that the model can generate extremely convincing samples of impersonated speech. It is even able to impersonate voices across different genders effectively. Results are qualitatively evaluated using standard procedures for evaluating synthesized voices.

Prior research that is of greatest relevance in this context relates to voice transformation, which deals with the specific problem of converting a source voice into a target one. Voice transformation has had a long history, and at the surface addresses some of the issues we mention. Conventionally, voice transformation modifies the instantaneous characteristics of a source signal, such as pitch [3] and spectral envelope. The strategies used range from simple codebook-based conversion [4] and minimum-mean-squared error linear estimators [5] to sophisticated neural network models [6]. While these methods are all frequently quite effective at transforming instantaneous characteristics of the signal, and can even map some prosodic cues, they are generally insufficient to capture unmeasurable, unquantifiable style in the more general sense of the word. When trained, they are heavily reliant on the availability of parallel recordings of the source and target speaker saying the same utterances, providing exact examples of what is considered ideal conversion. In most cases, in order to learn the voice conversion effectively, these recordings must also be perfectly time aligned, a requirement that is generally satisfied by time-warping the recordings to align them

to one another. Realization of the hard targets required to learn the conversion is not only unrealistic, the alignment required may also be fundamentally inappropriate when the objective is not to learn to perform wholesale conversion of voice, but only to transform style.

In this context, recent advances in the science of learning generative models provide us new directions. Rather than attempting to learn a mapping between parallel signals, the new models attempt to *discriminate* instead between data that do have the desired (identifiable but possibly unquantifiable) stylistic feature(s), and those that do not. Generators that attempt to produce data with any specific characteristic(s) must now learn to do so such that they "fool" the discriminator. Since the features are unquantifiable, the discriminator itself must, in fact also be learned. Both the generators and the discriminators are modeled by deep neural networks, which are known to be able to model any transformation with appropriate design and sufficient training data. Since the primary driver of the learning process is discrimination, parallel data such as those needed for conventional voice-conversion methods are not required.

These *Generative Adversarial Networks*, or GANs have been very successfully applied to a variety of problems in image generation [40], learning feature representations [41] and *style transfer* [42, 43, 44, 45, 46], wherein the algorithms involved result in fast and vivid generation of images of different artistic styles ranging from simple photographs to painting styles of selected artists. In our work, we harness the power of these models for the problem of style transfer in speech.

At the outset, we note that speech signals have several problems that are not inherent to images. Unlike images, speech sounds are not of fixed size (*i.e.*, not fixed in duration), and lose much of their stylistic characteristics when they are scaled down to be so. Generation of time-series data such as speech is also a more challenging problem compared to images. Naive implementations of the process may result in

generation of data that have lost linguistic, stylistic or even intelligible content. In this work, we propose multiple GAN models for the problem of voice transformation. Our models, and their corresponding learning algorithms, are designed to consider the specific challenges inherent in speech. Specifically, we show how, by appropriate choice of model structure and learning algorithm, and by introducing the appropriate discriminators in the GAN framework, *specific* characteristics of the voice might be retained without modifying others or losing linguistic content, in order to emulate different aspects of impersonation or voice mimicry.

In Section 3.1.1 we briefly outline the concept of GANs. In Section 3.1.2 we describe our designs of GANs for voice modifications. In Section 3.1.2 we present experimental evaluations of the proposed models and conclude with discussions in Section 3.1.2.

3.1.1 Generative Adversarial Networks (GANs)

In spite of their rather short history, GANs [40] are already quite well-known. We briefly summarize their key features here, in order to set the background for the rest of the chapter.

The basic GAN model

The Generative Adversarial Network is a generative model which, at its foundation, is a generative model for a data variable. The model is intended to generate samples that closely match draws from the actual distribution of the data. These models differ from conventional generative models in a fundamental way in the manner in which they are learned. Conventional generative models are trained through likelihood maximization criteria, such that some (empirical estimate of the) divergence measure between the synthetic distribution encoded by the generative models, and the true distribution of the data, is minimized. In contrast, GANs are trained discriminatively, such that

samples generated from the model cannot be distinguished from actual draws from the true distribution of the data.

Consider any random variable x with a probability distribution P_x that is unknown, but from which samples may be drawn. For instance, x may represent images of a particular class, samples of which may be readily available, but their actual distribution may be unknown. The GAN attempts to generate samples of x that are indistinguishable from actual samples drawn from the true distribution. The original GAN model [40] comprises a generator $G(z)$ and discriminator $D(x)$. The generator G takes as input a random variable z drawn from some standard probability distribution function P_z, *e.g.*, a standard Normal distribution, and produces an output vector x_z.

The discriminator $D(\cdot)$ attempts to discriminate between samples $x \sim P_x$ that are drawn from P_x, the true (but unknown) distribution we aim to model, and samples produced by the Generator G. Let T represent the event that a vector x was drawn from P_x. The discriminator attempts to compute the *a posteriori* probability of T, *i.e.*, $D(x) = P(T|x)$.

To train GAN, we attempt to learn G such that $D(x_z)$, the score output by the discriminator in response to productions by G is maximized (*i.e.*, G "fools" the discriminator). At the same time we attempt to learn D such that $D(x_z)$ is minimized, while also maximizing $D(x)$ for any $x \sim P_x$. All of these objectives can be concurrently achieved through the following optimization:

$$\min_G \max_D E_{x \sim P_x}[\log D(x)] + E_{z \sim P_z}[\log(1 - D(x_z))]$$

The GAN training framework is illustrated in Figure 3-1.

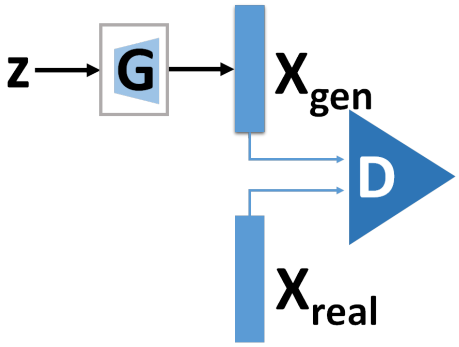


Figure 3-1: The original GAN model

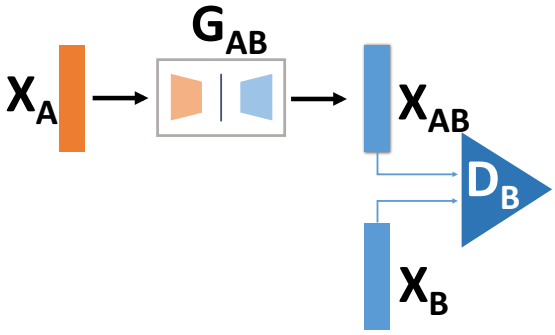


Figure 3-2: Style transfer by GAN

GANs for style transfer

The basic GAN has been extended in a number of ways in the literature [42, 43, 44, 45, 46], particularly in the context of style transfer among images, *e.g.*, as in Figure 3-2. The common underlying denominator in all of these models is that an input data instance (usually an image) x_A drawn from a distribution P_A is *transformed* to an instance x_{AB} by a generator (more aptly called a "transformer"), G_{AB}. The aim of the transformer is to convert x_A into the style of the variable x_B which natively occurs with the distribution P_B.

The discriminator D_B attempts to distinguish between genuine draws of x_B from P_B and instances x_{AB} obtained by transforming draws of x_A from P_A. The actual

optimization is achieved as follows. We define

$$L_G = E_{x_A \sim P_A}[\log(1 - D_B(x_{AB}))]$$
$$L_D = -E_{x_B \sim P_B}[\log D_B(x_B)] - E_{x_A \sim P_A}[\log(1 - D_B(x_{AB})] \tag{3.1}$$

To train the GAN, its two components are alternately updated by minimizing the two losses in Equation 3.1. The generator G is updated by minimizing the "generator loss" L_G, while the discriminator is updated to minimize the "discriminator loss" L_D.

Our work is however more directly based on the "DiscoGAN" model [42], shown in Figure 3-3. The DiscoGAN is a symmetric model which attempts to transform two categories of data, A and B, into each other. The DiscoGAN includes two generators (more aptly called "transformers") G_{AB} and G_{BA}. G_{AB} attempts to transform any draw x_A from the distribution P_A of A into $x_{AB} = G_{AB}(x_A)$, such that x_{AB} is indistinguishable from draws x_B from the distribution P_B of B. G_{BA} does the reverse.

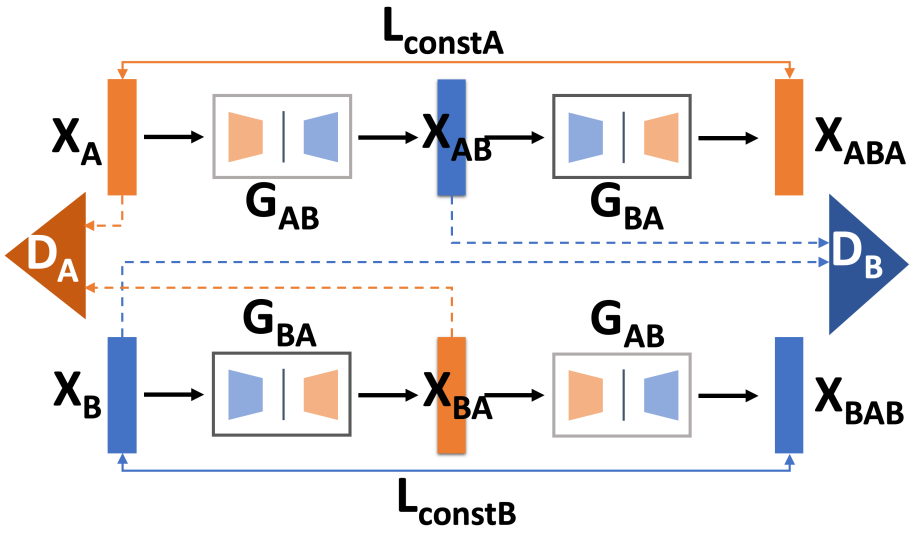


Figure 3-3: The DiscoGAN model

The DiscoGAN model also includes two discriminators, D_A and D_B. D_A attempts

to discriminate between actual draws from P_A and draws from P_B that have been transformed by G_{BA}, and D_B performs the analogous operations for draws from P_B. The generators and discriminators must all be jointly trained.

The training process for the DiscoGAN is similar to that for the model in Figure 3-2, with one significant modification: in addition to the losses that emphasize the competition between the generators and the discriminators, we now include the requirement that G_{AB} and G_{BA} must be inverses of each other to the extent possible, *i.e.*, for any x_A from A, $x_{ABA} = G_{BA}(G_{AB}(x_A))$ must be close to the original x_A, and similarly for any x_B from B, $x_{BAB} = G_{AB}(G_{BA}(x_B))$ must be close to the original x_B. This requirement is encoded through two reconstruction losses L_{CONST_A} and L_{CONST_B} where

$$L_{CONST_A} = d(G_{BA}(G_{AB}(x_A)), x_A) \tag{3.2}$$

and L_{CONST_B} is symmetrically defined. The generator loss for G_{AB} is defined as:

$$L_{GAN_{AB}} = L_{CONST_A} + L_{G_B} \tag{3.3}$$

where L_{G_B} is defined as in Equation 3.1. We define the generator loss for G_{AB} in a symmetric manner. The overall generator loss is $L_G = L_{GAN_{AB}} + L_{GAN_{BA}}$. The discriminator loss L_D is defined as $L_D = L_{D_A} + L_{D_B}$, L_{D_A} and L_{D_B} are defined as in Equation 3.1. Finally, in the implementation of DiscoGAN [42], a *feature loss* is also added to compare the feature similarity between the generated data and the real data. As before, the generators and discriminators are trained by alternate minimization of the generator and discriminator losses.

3.1.2 GANs for voice mimicry

The DiscoGAN was originally designed to transform style in images. In order to apply the model to speech, we first convert it to an invertible, picture-like representation,

namely a spectrogram. We operate primarily on the *magnitude* spectrogram, retaining the phase of input signals to be transformed, to recreate the transformed signals from the transformed magnitude spectrogram.

In order to apply this to voice transformation, we must make several key modifications to the DiscoGAN model. Firstly, the original DiscoGAN was designed to operate on images of fixed size. For it to work with inherently variable-sized speech signals, this constraint must be relaxed in its new design. Secondly, it is important to ensure that the *linguistic* information in the speech signal is not lost, even through the signal itself is modified. Sufficient constraints must be added to the model for this. Finally, since our objective is to modify specific aspects of the speech, *e.g.*, style, we must add extra components to our model to achieve this. We call our model, which incorporates all these modifications, the *VoiceGAN*.

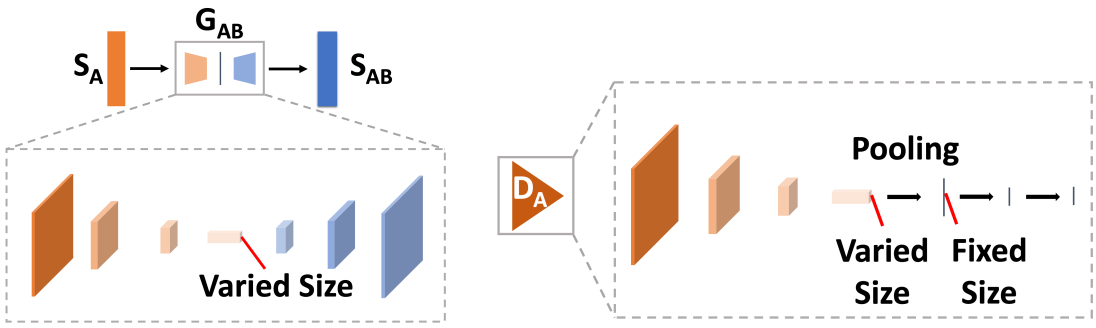
Retaining Linguistic Information. Linguistic information is encoded largely in the details of the spectral envelope. To ensure that this is retained, we modify our reconstruction loss as:

$$L_{CONST_A} = \alpha d(x_{ABA}, x_A) + \beta d(x_{AB}, x_A) \tag{3.4}$$

Here, the term $d(x_{AB}, x_A)$ attempts to retain the *structure* of x_A even after it has been converted to x_{AB}. Careful choice of α and β ensures both, accurate reconversion and retention of linguistic information, after conversion to x_{AB}.

Variable-length Input Generator and Discriminator

To account for the fact that unlike images, speech signals are of variable length that cannot be scaled up or down, we must make modifications to the generators and discriminators. The modified structures are shown in Figure 3-4. Figure 3-4 (a) shows the structure of the original generator in DiscoGAN. Based on its fully



(a) Generator G_{AB} (in box) (b) Discriminator D_A (in box)

Figure 3-4: Visualization of Generator G_A and Discriminator D_A. Architectures of other counterparts are similar in structure. The number of convolutional layers is larger in the actual implementation.

convolutional structure, it can handle variable length inputs. Figure 3-4 (b), we shows the architectural details for our proposed discriminator in VoiceGAN. In this, an adaptive pooling layer is added after the CNN layers, and before the fully connected layer. It includes channel-wise pooling in which each channel's feature map is pooled into a single element. This converts any variable-sized feature map into a vector of a fixed number of dimensions, with as many components as the number of channels.

Style Embedding Model (D_S). In addition to the discriminator that distinguishes between the generated data and real data, we add a second type of discriminator to our model to further extract the target *style* information from input data and to make sure that the generated data still has this style information embedded in it. To achieve this, we include a discriminator D_S that is similar in architecture to that in Figure 3-5.

The discriminator D_S determines if the original and transformed signals match the desired style. To do so, we introduce the following style loss:

$$\begin{aligned} L_{D_{STYLE-A}} = d(D_S(x_A), label_A) + d(D_S(x_{AB}), label_B) \\ + d(D_S(x_{ABA}), label_A) \end{aligned} \tag{3.5}$$

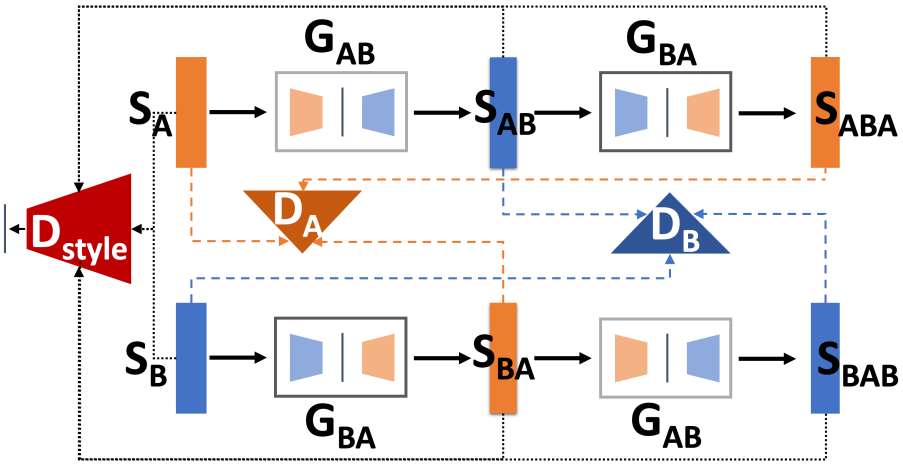


Figure 3-5: The proposed VoiceGAN Model

$$L_{D_{STYLE}} = L_{D_{STYLE-A}} + L_{D_{STYLE-B}} \tag{3.6}$$

Note that the style loss could include multiple discriminators for multiple aspects of style.

Total Loss Our final training objectives to be minimized for the generator and discriminator are represented by L_G and L_D respectively as follows:

$$\begin{aligned} L_G &= L_{GAN_{AB}} + L_{GAN_{BA}} \\ &= L_{G_B} + L_{CONST_A} + L_{G_A} + L_{CONST_B} \end{aligned} \tag{3.7}$$

$$L_D = L_{D_A} + L_{D_B} + L_{D_{STYLE}} \tag{3.8}$$

Experiments and Results

We use the TIDIGITS [47] dataset. This dataset comprises a total of 326 speakers: 111 men, 114 women, 50 boys and 51 girls. Each speaker reads 77 digit sentences. The sampling rate of the audio is 16000 Hz. We chose to use this database due to

its relatively simple linguistic content. For the purpose of demonstration, we choose an unquantifiable, but identifiable characteristic: gender. Our goal then is to show that these data can be used to *learn* to convert the gender of a speaker's voice. In the discussion below, therefore, "style" refers to gender. We note that any other characteristic may have been similarly chosen.

Model implementation

The model architecture is that of the VoiceGAN described above. The generator network in the model comprises a 6-layer CNN encoder and a 6-layer transposed CNN decoder. The discriminator network comprises a 7-layer CNN with adaptive pooling. We employ batch normalization [48] and leaky ReLU activations [49] in both the networks. The number of filters in each layer is an increasing power of 2 (32, 64, 128). When training the networks, a smoothness constraint, comprising the cumulative first order difference between adjacent columns in the spectrogram, is added to the loss to enhance the temporal continuity of the generated spectrogram. Results are available at [50].

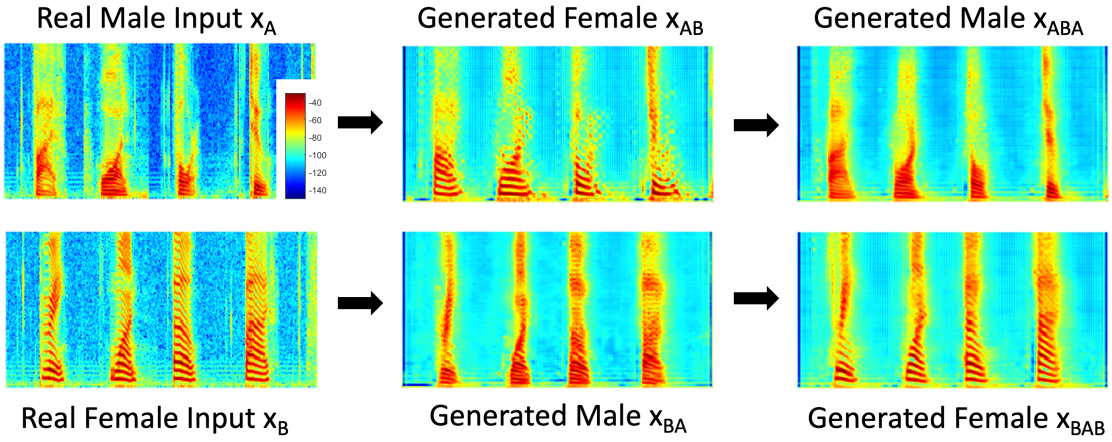


Figure 3-6: Visualization of the spectrograms through the voice transformation. The contexts of these utterances are a speaker saying "3 1 oh 5" (first row) and "5 1 4 2" (second row). For each spectrogram, frequencies on the y-axis range from 0-4 kHz. You could observe the fine nuances in the spectrograms are consistent for the same gender across the transformations.

Table 3.1: NIST STNR test

Data (use GL-method)	A (dB)	B (dB)
Original signal	55.60±4.97	52.91±3.58
X_A and X_B	54.97±6.28	52.15±3.70
X_{AB} and X_{BA}	49.64±1.80	49.92±4.36
X_{ABA} and X_{BAB}	53.58±2.69	50.05±2.12

Quality evaluation of generated results

We use an independently-trained CNN-based classifier to predict the style of our generated data. The classifier was trained on 800 utterances from speakers of both genders. The results show that 100% of the generated data are classified as the target speaker's style, which indicates that our VoiceGAN network achieves good style transfer performance.

To evaluate the quality of our generated speech signal [51], we also conduct a signal-to-noise (SNR) ratio test using the standard NIST STNR method and the WADA SNR method [52]. The results are shown in Table 2. For each data class, we randomly select 40 samples from our test dataset (20 for each speaker) and compute the mean and variance of the generated results. The WADA test results are all around 100 dB since our generated noise is not well-modeled by Gaussian noise. The STNR test results show that our generated data is of good quality. For evaluation, the time-domain signal is reconstructed from the generated spectrogram using the Griffin-Lim method, which is based on an iterative procedure that minimizes the mean square error between the modified magnitude spectrogram and the actual signal's spectrogram. Details of this method are explained in [53]. We find that the Griffin-Lim method does not reduce the voice quality to any significant degree.

Conclusions

The VoiceGAN model is observably able to transfer style from one speaker to another. As proposed however, this model remains vanilla and many extensions are possible.

The method can be easily extended to other stylistic features that may be identified. In principle, while longer-term prosodic-level style features may also be transferred, simple binary discriminators may no longer be useful for such characteristics. More continuous-valued discrimination may be required. We have not verified if multiple style aspects may be *concurrently* modified. These remain areas of ongoing research. In preliminary experiments we have verified that even *linguistic* content may be modified if we so choose; however doing so in a measurable and controlled manner is a challenge that remains to be addressed. Future works in voice conversion related to our proposed the VoiceGAN model [54] have continued to incorporate the most relevant innovations in the area of adversarial modeling [55, 56, 57].

3.2 Speech synthesis

Another approach to generate deepfake speech is to synthesize the target utterance from the text or other language features such as symbolic linguistic representations. To obtain the target speaker's identity, the synthesized speech should have the voice characteristics and spoken styles of the target.

For speech synthesis, especially text-to-speech (TTS) synthesis, the TTS models are usually pretrained with a large dataset with a fixed set of speakers. The learned parameters are the ones adapted to the speakers inside the training data so that during the inference phase, the synthesized utterance has the training speaker's voice. In this setting, it is not easy to deepfake the target's speaker's voice without a large dataset of their voice recordings. A number of approaches and methods have hence been proposed to reduce data requirements for imitating a target speaker's voice and speaking styles [58, 39, 59, 58, 60, 61].

To have a better understanding of deep-fake synthesis through text-to-speech models, we have done two works that are the state-of-the-art of TTS with prosody and

styles at the time of this writing.

The first one is for prosody transfer and the second one is for specific-style-matched synthesis. Both of them can be used to generate deepfake speech that capture the styles/emotions of the target. Through the investigation of these two works, we obtains insights for better detection of deep-fake speech.

3.2.1 Prosody transfer

High-quality text-to-speech (TTS) synthesis has remained a challenging research topic for years. Pushing the edge of the general naturalness of the synthesized utterance, several state-of-the-art models such as Tacotron and DeepVoice3 achieve excellent results in improving the quality of synthesized speech. To aim at more realistic speech synthesis, prosody-flexible TTS, also called expressive TTS has recently become a topic of significant research. For example, Google has proposed an expressive TTS framework to successfully learn a reference utterance's prosody and transfer it to a new utterance synthesized by the system. In this session, we propose a prosody transfer text-to-speech synthesis model. Our work is implemented based on the end-to-end CNN block-based model of Baidu's DeepVoice3 (DV3). Different from former models, in our work, we use a joint-attention learning process of the reference prosody and text. This comparatively simpler model can learn the reference input's prosody along with the text input. A token table and weights are also learned with the reference input to factorize the possible styles in an unsupervised manner. The results show our model can successfully factorize the reference prosodies to represent characteristics of different speakers and styles, under unsupervised learning from the training data.

Introduction

Text-to-speech (TTS) research aims to develop models that can produce natural-sounding synthesized utterances, given some text as input. It has been the focus of

most recent research to improve the "naturalness" of the produced speech.

First, we would like to define what the term "natural" means in TTS synthesis. The naturalness of speech is usually hard to measure directly and quantitatively by an algorithm; instead it is usually quantified through aggregated subjective opinion of human experts using their own years of experience in speech communication. The "naturalness" of a speech utterance can be can be construed as having the right linguistic content, correct phonemic pronunciation, clear speech quality as well as a good speech style or prosody.

To achieve this goal of improving naturalness in TTS synthesis, there are many successful models that have been proposed recently, such as the Google's Tacotron model [62, 63] and Baidu's DeepVoice models [64, 65, 66]. These high-performance modern TTS systems compute outputs based on the statistics of the training data, which usually lead to high-quality clean speech with average speaker characteristics and prosody style.

However, this still remains a very interesting challenge for TTS synthesis: apart from learning the average speaker characteristics and styles, a more controllable synthesis model that is flexible enough to learn and produce the prosodic styles of different speakers is in high demand. One approach to enrich the ability of the TTS system in this respect is to extend its capabilities from single-speaker to multi-speaker synthesis. There have been several important advances in the area of multi-speaker TTS, a few of the most notable ones being DeepVoice2 [65] and VoiceCloning [67] as well as [68]. In multi-speaker TTS training, a *speaker-encoder* is included in the system. This can take a speaker's identity as input, and use it to associate the speaker's training data with features learned from specific speaker-ID embeddings, so that the network can work on speaker-customized features. In the test phase, given a speaker ID, an utterance with the corresponding speaker's voice characteristics could be synthesized using the corresponding embeddings.

A second approach – that of expressive TTS – is more direct. Specific features that represent certain "prosody" are learned in the training phase, and used in the synthesis phase. The term prosody here represents the remaining variation in speech signals after discounting for the variations due to phonetics, speaker identity, and channel effects. It usually includes many characteristics such as pitch, stress, breaks, and rhythm [2, 59]. Based on the state-of-the-art TTS system Tacotron, some impressive Expressive TTS models [2, 58, 59] have been proposed, which can model the prosodies and styles in an unsupervised manner with no explicit prosody labels in the training dataset.

Related works

In this section, we introduce some prior research that is closely related to the subject of this session.

Baidu's Deep Voice 3 is a state-of-the-art, fully-convolutional, attention based TTS system. It achieves impressive high-fidelity speech synthesis. As shown in Figure3-7, the system is composed of an *encoder*, a *decoder* and a *converter*. The encoder is a fully-convolutional network that takes textual features as input, and outputs learned text representations that serve as input to the decoder. The decoder is also a fully-convolutional network that converts the learned text representations (the input from the encoder), to a low-dimensional audio representation, utilizing an attention mechanism in an autoregressive setting. The converter is a post-processing network that converts the output of the decoder to the final output that can be fed into a voice vocoder. If there are multiple speakers in the recordings used for training, the corresponding speaker embeddings will be inserted into the encoder, decoder and converter to successfully help the entire network learn the differences among the speakers. The basic framework used in our work is implemented based on the same strategy as in DeepVoice3.

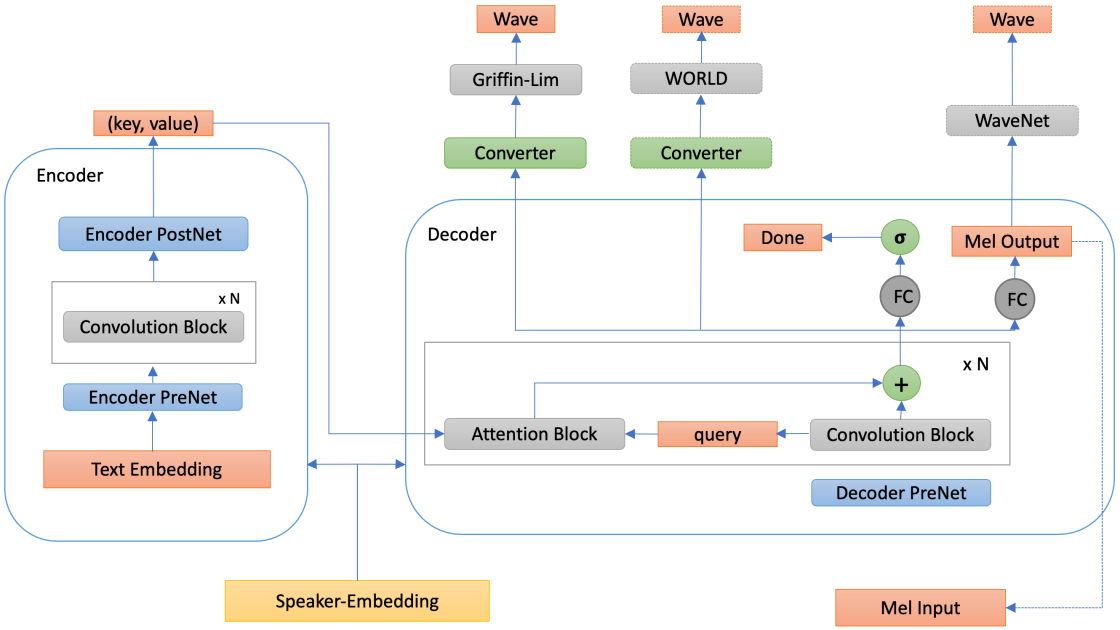


Figure 3-7: A simplified flowchart of Baidu DeepVoice3 system

There are several recent approaches that successfully represent a wide range of speaking styles and try to disentangle prosody factors from among those. One example is [59], which is an architecture based on the Tacotron that can learn latent representations of prosody from reference utterances. The conditioning Tacotron can then transfer the prosody from the reference to the synthesized audio using the learned prosody embeddings of the reference audio. This achieves high-performance prosody transfer from the reference audio to the synthesized utterance, with the successful transfer of finer details of prosody, as evidenced by their recent demonstration. In their work, a reference encoder is built to extract prosody embeddings from spectrographic slices of the reference audio. The prosody is represented by fixed-length embeddings, which are observed to be more robust to text and speaker perturbations.

To account for the undetermined prosody features, an improvement over this approach is proposed in the Style Tokens (GSTs) method [2]. The GST method is built up from the above mentioned prosody-transfer TTS network proposed by Google. In this, the reference encoder is similar to the former prosody transfer network's reference

encoder – it can compress the prosody of reference inputs of variable lengths to a fixed-length vector.

A schematic diagram of the GST approach is shown in Fig. 3-8. As shown in this figure, the reference embedding is exacted from the reference encoder, which is the same as the one in the former prosody transfer network. To disentangle and factorize the finer details of prosody, a novel *style token layer* is proposed in the GST. The reference embeddings can be given as input to this layer. An attention module in the GST network treats the reference embeddings as a query vector, learning the similarity between the reference embeddings (query) and each token in the token bank (responses). The token banks contain a set of randomly initialized embeddings, which are called *global style tokens* (GSTs). The output of the attention module is a set of weights that represent the similarity measurement of each token to the reference embedding. The weight matrix multiplied by the GSTs is then a weighted sum of all tokens, and called the *style embedding*.

In [2], a content-based attention model is used as a mechanism to measure the similarity between the global token and reference embeddings. In the experiments reported in [2], it was found that using a multi-head attention [69] model can significantly improve the style transfer performance. In the two works mentioned above, reference audio is needed during inference to render the reference prosody's style to the inferred audio. As an improvement over this, a Text-Predicted Global Style Token (TP-GST) is proposed in [61]. The TP-GST model can learn to predict possible styles from text alone, so it requires no explicit labels during training and no auxiliary inputs for inference.

The model

Our prosody transfer TTS system was built up from the open-source Baidu's DV3 system. The model is shown in Figure3-9. Building upon the DV3, we additionally

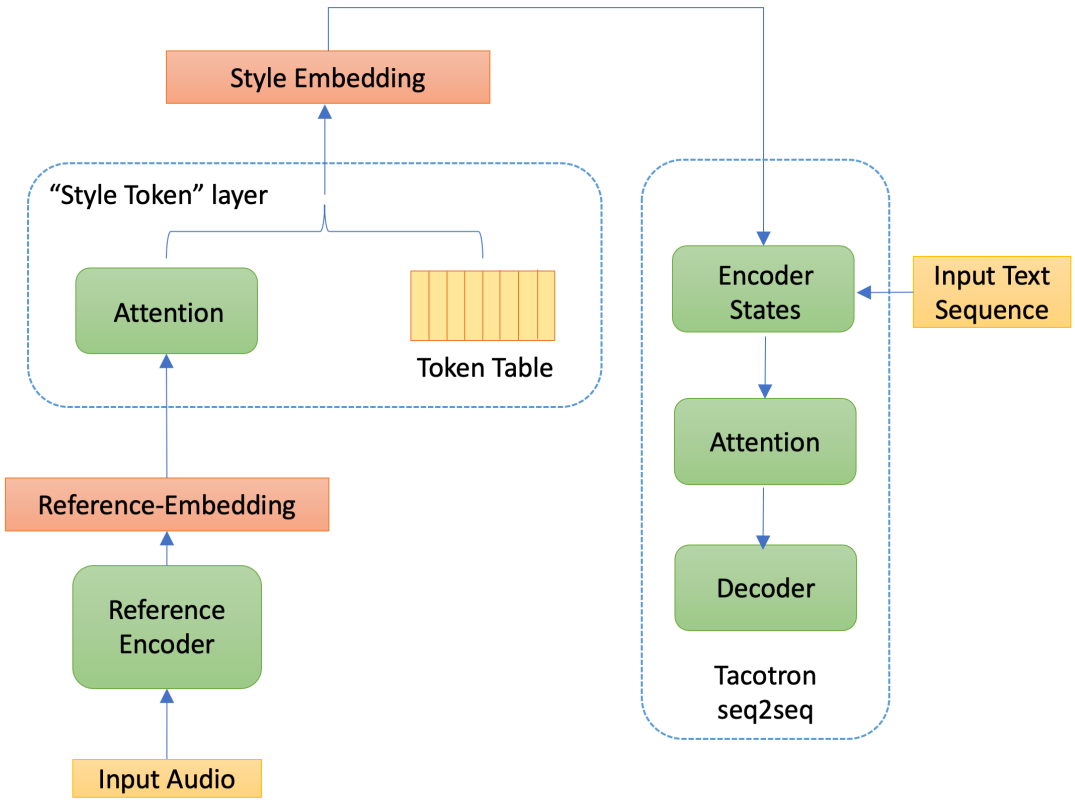


Figure 3-8: Training phase of style tokens [2]

incorporated a reference encoder into the framework. The reference encoder learns the extracted weights directly from the reference audio. The weights matrix is then directly multiplied with the randomly initialized token table to give the combined tokens as reference embeddings. In our model, there is no explicit attention module used to learn the similarity between the reference audio's features and the global tokens' table. In contrast to Google's GST approach, the predicted weights are directly extracted from the reference utterance (input) and combined with the global token table to give a reference embedding for further use.

The learned reference embedding is then forwarded to the text encoder in the Encoder PreNet and Convolution Blocks, so that a jointly learned (key, value) pair based on the text and reference style is fed into the attention block of the decoder.

In contrast to the content-based attention module utilized in [2], which learns a

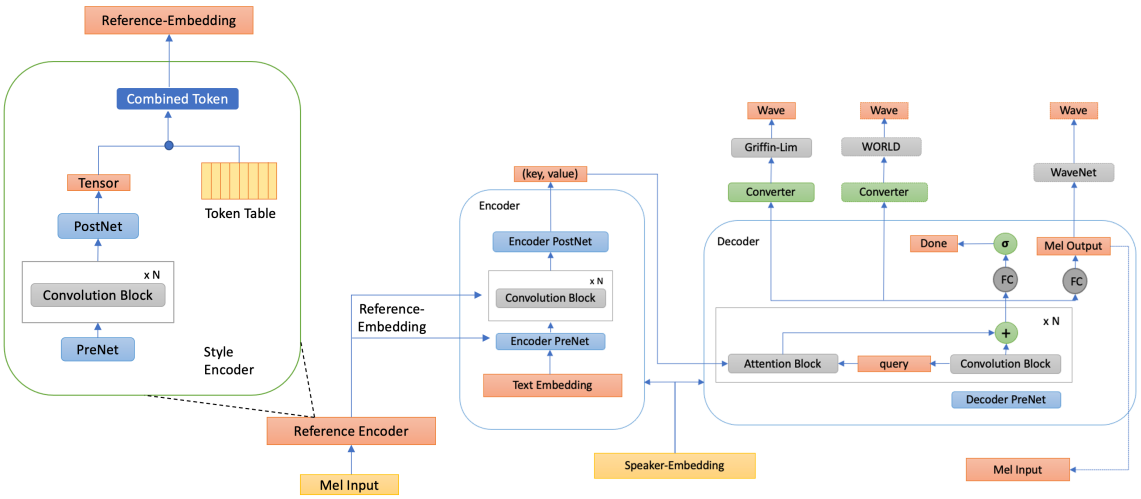


Figure 3-9: Our proposed reference encoder model (left zoomed in)

similarity between the reference embedding and each token, a joint-attention learning mechanism is used to simplify the learning process in our approach. The reference embedding is learned directly as a combination of weights that are exacted from the reference utterance, with a bank of randomly initialized token tables, so that the reference embedding itself becomes a combination of the token table and predicted weights. This learned embedding is then directly concatenated with the features from the text embedding module to learn the (key, value) pair for the attention model in the decoder. Thus the attention block in the decoder learns the attention based on the combination of text and reference embeddings.

In this mechanism, the prosody variation is learned in an unsupervised manner by the network during training, and the weights from the reference are predicted in order to combine them with the token table to give combined tokens as reference features. Both the parameters used to predict the weights and the global token table values are fine-tuned during back-propagation to accomplish the goal of extracting and representing the reference utterance's prosody characteristics.

Experiments

There are two datasets used in the training of this network, respectively, the VCTK dataset [70] and the Blizzard dataset [71]. VCTK dataset is a multi-speakers dataset containing 108 native English speakers (who have corresponding text information) with different accents and a total duration of around 44 hours. Each speaker reads out about 400 sentences, most of which were selected from a newspaper plus the Rainbow Passage and an elicitation paragraph intended to identify the speaker's accent [70]. Blizzard dataset is a dataset very suitable for expressive TTS research. It contains approximately 300 hours of recordings of audiobook data provided by The Voice Factory, from a single female speaker. Since the professional female speaker recorded the audiobook reading under an quiet controlled environment. The recordings are very clean and clear. Furthermore, the emotional and prosody variations that existed in the recordings are ideally for prosody study in expressive TTS research.

The model is implemented based on the open-source implementation [72] of Baidu's DeepVoice3 system. In the implementation, when using VCTK dataset, speaker-ID is assigned to the same value to force speaker characteristics learned to the token representations.

The reference encoder is composed of a PreNet and PostNet with similar structures as in the Text encoder and six layers of CNN blocks. The Token table is randomly initialized as a 10×100 matrix tensor with ten as the token number and one hundred as the token dimensions.

Results and Discussion

To verify the effectiveness of our system, we first train the model using VCTK dataset without feeding different speaker IDs into the system. Therefore, the reference token table is expected to learn different speakers' characteristics from the strongest ··styles‘' represented in this VCTK dataset. During inference phase, we directly condition the

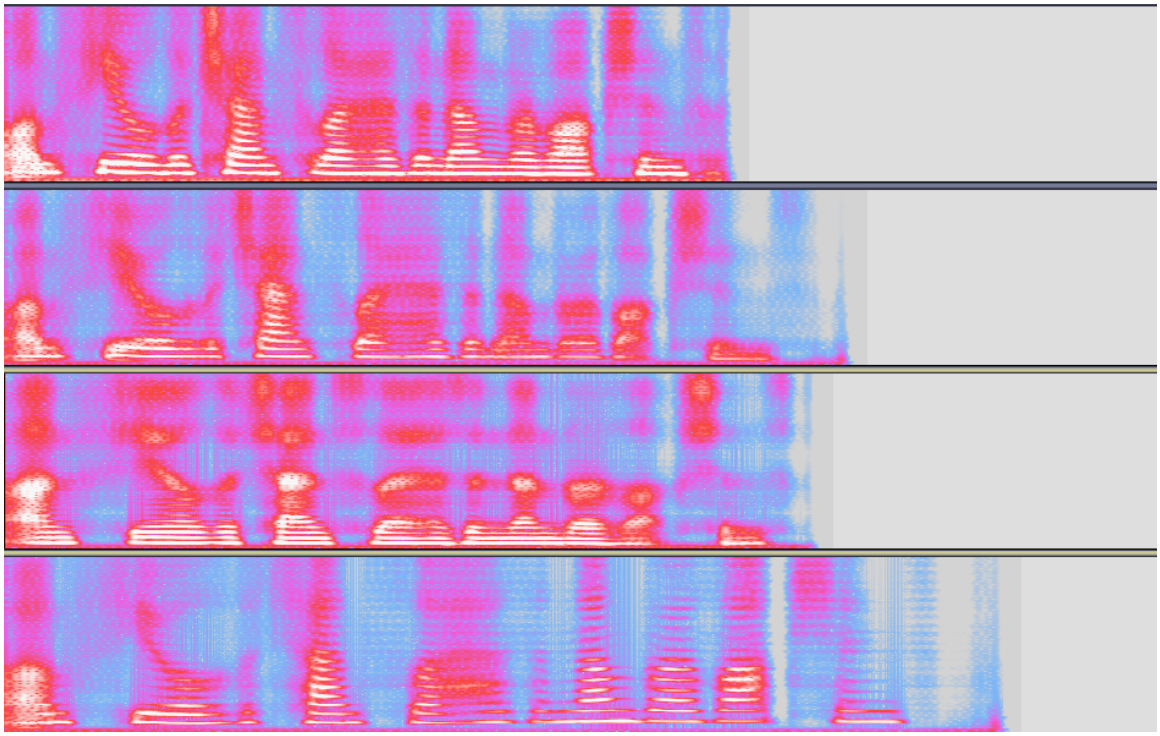


Figure 3-10: The same utterances inferenced using four different tokens learned through the TTS training with VCTK dataset. The utterance is "I've felt the chance that I have a number of options". Row 1 to 4 are the synthesized results using token 1 to 4. X axis is the time frame and y axis is the frequency from 0 - 4k.

network on certain tokens as the reference embedding. Therefore, the synthesized utterance will represent the style of that token and reflect the learning ability of the token table.

The results showed that when we assign a specific token number during the reference, different tokens can give the voice of different speakers. This verifies the effectiveness of our system. The spectrograms of the results are shown in Figure 3-10. The pitch and prosody of each utterance is different, synthesized from the reference embedding of different tokens, as shown in Figure 3-12.

With the assigned one-hot vector as token weights, we can explore a certain token's impact on the synthesized utterance. Figure 3-11 is the linear spectrogram visualization under audacity of the same utterance "Just recovered a fumble on ensuing kickoff." synthesized using different tokens. The prosodies of the synthesized

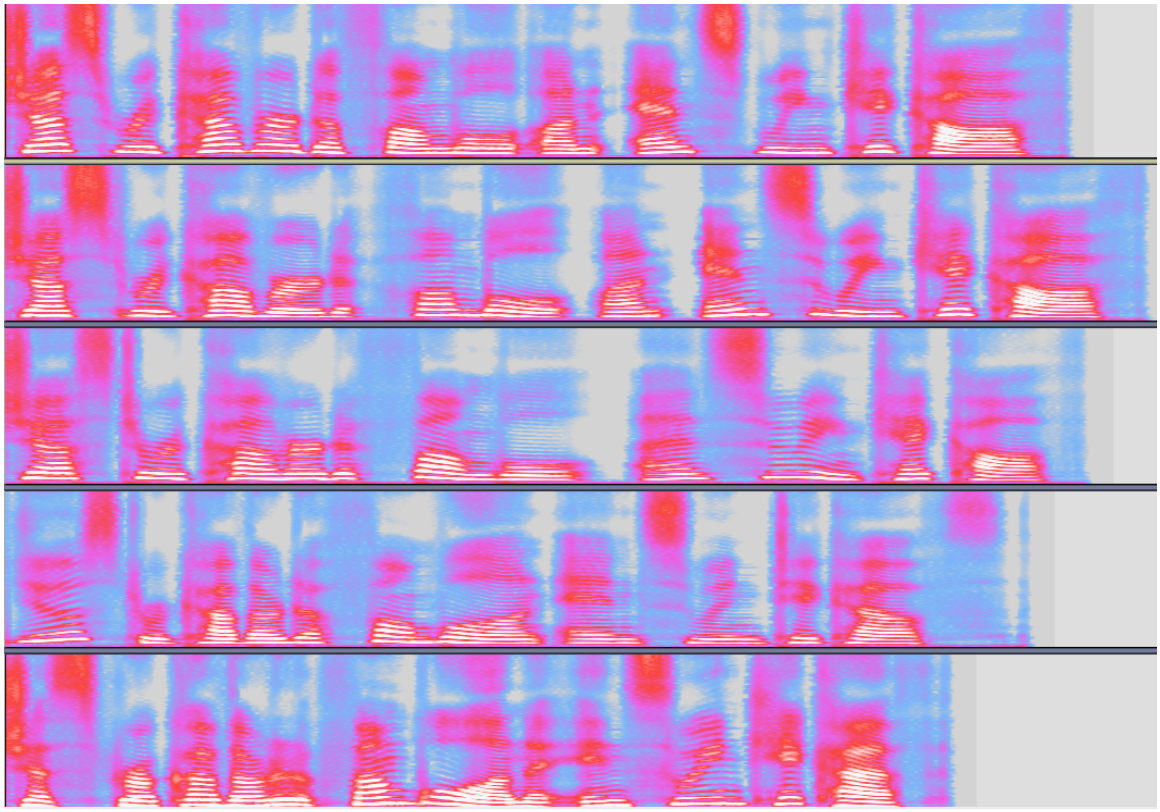


Figure 3-11: The same utterances inferenced using token No.1-5. The utterance is "Just recovered a fumble on ensuing kickoff". X-axis is time frame and y-axis is the frequency from 0 - 4k.

utterance are different. The prosodies varies from neutral to fast-pace, from calm and relaxing to certain emphasizing on specific words.

Figure 3-14 shows that the F0 contours of synthesized utterances follow a clear relative trend among respective tokens, even when the text contents are totally different. Given the representation of reference audio, we could learn the prosody of the reference and transfer it to the synthesized audio.

Figure 3-13 is a spectrograms visualization of the reference prosody to the synthesized one. The uppermost utterance is an inference result without a specific prosody reference, giving only text as inference input. The second row is the spectrogram of prosody transfer synthesized result. The word "anymore" is stretched, and the speaking pace is comparatively slow, capturing the reference utterance's prosody.

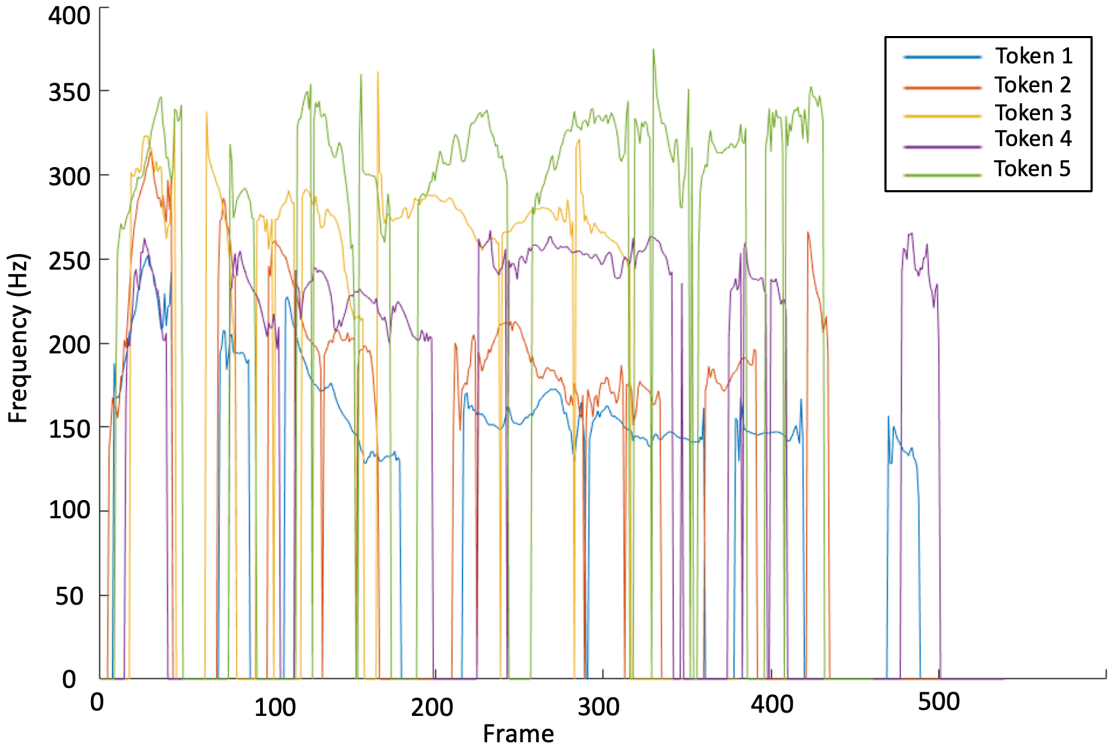


Figure 3-12: F0 visualization of synthesized utterance using different tokens. The utterance is "Just recovered a fumble on ensuing kickoff". Different tokens give different prosody synthesis results, as the F0 sequence shown, for the same text content.

The audio demos can be found in this demo page [73].

Conclusions

A reference encoder is built based on the DeepVoice3 model, which can learn the reference's prosody successfully. In this proposed model, a simple joint-attention learning method is used to combine the reference embedding with the text embedding in the decoder attention learning. The results show that our model can successfully factorize different prosodies by learning different prosodies using style tokens. As a result, the reference encoder can learn the reference utterance's prosody to transfer to the synthesized utterance during inference. In future works, a more detailed exploration of the attention model can be studied to further improve the prosody transfer performance. Furthermore, there are many promising potentials that are waiting to

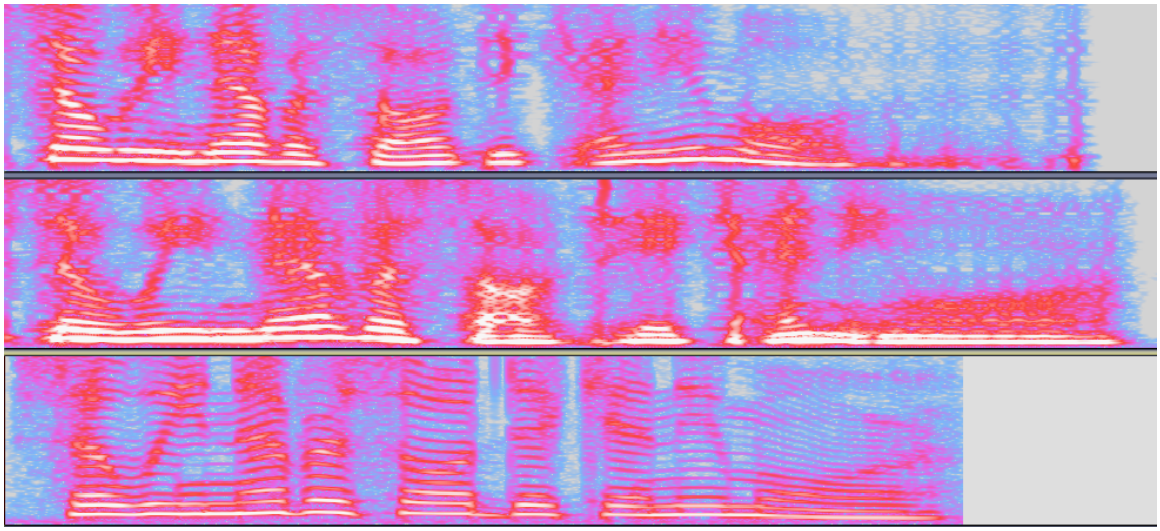
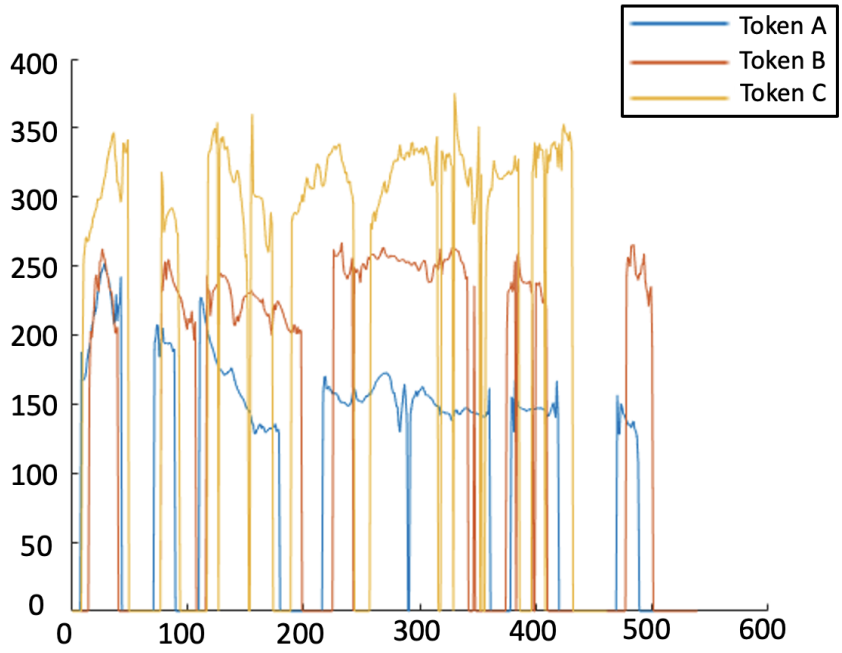


Figure 3-13: The same utterances syntheized using different prosodies. The utterance is "So we never saw Dick anymore". The uppermost one is a neutral prosody synthesized result. The middle utterance is slow-pace and has emphasis on the word "anymore", the same as the bottom reference. X axis is time frame and y axis is the frequency from 0 - 4k.

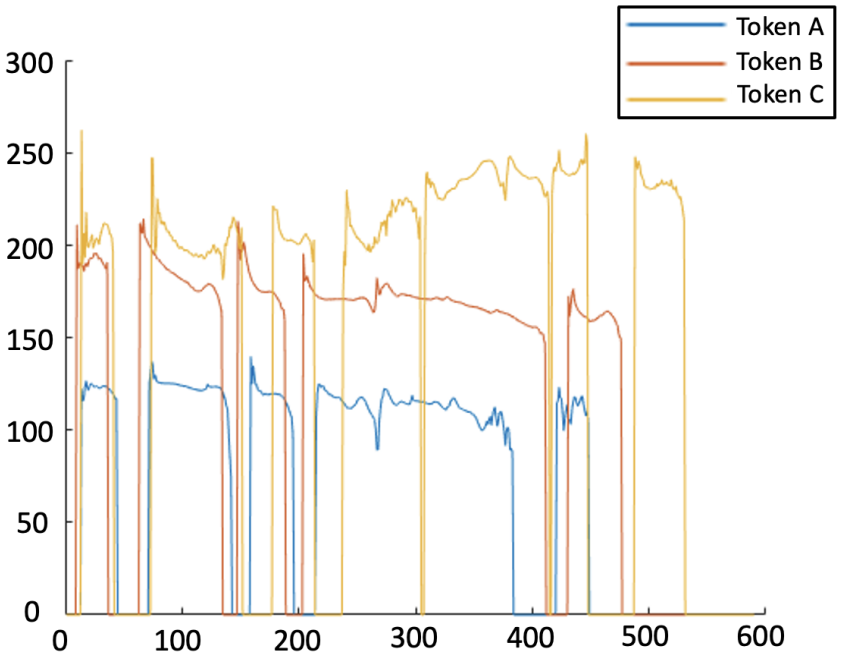
be studied, such as denoising effects of token representation and more explicit style disentangling in training.

3.2.2 Voice synthesis with style characteristics

While modern TTS technologies have made significant advancements in audio quality, there is still a lack of behavior naturalness compared to conversing with people. We propose a style-embedded TTS system that generates styled responses based on the speech query style. To achieve this, the system includes a style extraction model that extracts a style embedding from the speech query, which is then used by the TTS to produce a matching response. We faced two main challenges: 1) only a small portion of the TTS training dataset has style labels, which is needed to train a multi-style TTS that respects different style embeddings during inference. 2) The TTS system and the style extraction model have disjoint training datasets. We need consistent style labels across these two datasets so that the TTS can learn to respect the labels produced



(a) Sentence A



(b) Sentence B

Figure 3-14: Visualization of F0 of two different utterances. Sentence A: "Just recovered a fumble on ensuing kickoff."; sentence B: "I've felt the chance that I have a number of options". They are synthesized using three tokens A, B, and C. Independent of the text content, the same taken can preserve similar comparative F0 trends, respectively.

by the style extraction model during inference. To solve these, we adopted a semi-supervised approach that uses the style extraction model to create style labels for the TTS dataset and applied transfer learning to learn the style embedding jointly. Our experiment results show user preference for the styled TTS responses and demonstrate the style-embedded TTS system's capability of mimicking the speech query style.

Introduction

With increasing interest in interactive speech systems such as voice assistants, there is an increased demand for human-like text-to-speech (TTS) systems. While recent technology advancements in speech synthesis have achieved human-like audio quality [74, 75, 69, 31], the TTS's speaking style does not mimic the naturalness and expressiveness as in human conversations, because conventional speech interfaces respond to input speech queries with default speaking style learned from the TTS training dataset. To make the TTS more interactive, the TTS's response should vary depending on the context and the speaking style of the input speech query. For example, when the user is speaking fast and rushing out the door in the morning, the TTS would match the hurried pace; and when the user is in a quiet place and is speaking softly, the TTS would respond with a soft and quiet voice. By detecting the input speech query's style and generating response accordingly, TTS can provide a more natural and customized user experience. One way to achieve this interaction is to incorporate two key components: a style extraction model that detects the speaking style of the input speech query and generates a style embedding, and a multi-style TTS system that can synthesize styled speech with respect to different style embedding inputs.

The challenge lies in jointly training the style extraction model and the multi-style TTS so that the style embeddings generated by the style extraction model can be genuinely respected by the TTS, even though the two components are trained with different datasets and style labels. In this paper, the TTS dataset is a commissioned

dataset recorded with professional voice talents. Only a small part of the TTS dataset has style labels. For the style extraction model, we make use of the external IEMOCAP dataset. These two datasets have different style labels. In order to achieve consistent labels between TTS training data and unseen queries, we incorporated both IEMOCAP dataset and a small portion of the labeled TTS dataset in the style classifier model's training.

We first train a multi-modal style classifier using the IEMOCAP dataset with the model described in [76]. Taking the softmax layer of the style classifier as style embedding, the classifier serves as a style embedding extraction model. This model is applied to the unlabelled TTS dataset to generate the style embeddings in a semi-supervised fashion. By using the style embedding as additional auxiliary features for the TTS system, we could train a controllable multi-style TTS system that learns to respect given target styles. During speech synthesis, style embedding is first extracted from the input speech query and then fed into the TTS system to produce response in matching styles. In summary, we developed an interactive multi-style TTS system that could lead to natural, expressive human-machine speech interactions. The multi-style TTS system is evaluated using comprehensive subjective experiments.

Related work

Emotion recognition

Early approaches to emotion recognition have mostly been inspired by studies in psychology [77, 78]. Recently, deep neural networks (DNNs) have first been shown to effectively learn high-level representations for utterance-level emotion recognition [79]. Trigeorgis *et al.* (2016) further applied convolutional neural networks (CNNs) to model context-aware emotion-relevant features, which are then combined with long term-short memory (LSTM) networks for end-to-end emotion modeling [80]. Emotion is generally expressed through multi-modal behavior, including speech, language,

body gestures, or facial expressions. Thus, emotion recognition is often formulated as a classification problem of utterances using these multi-modal signals. [81] proposed a multi-modal dual recurrent encoder to simultaneously model the dynamics of both text and audio signals within an utterance to predict emotion classes. This architecture has achieved state-of-the-art performance on the IEMOCAP[82] dataset, which is a multi-modal emotion dataset and has been widely used in the affective computing community.

Expressive TTS

One popular topic in the recent research on TTS is expressive TTS. A number of approaches have historically been proposed for expressive TTS, from HMM-based synthesis using a control vector for modeling style [30, 83, 84], to the state-of-the-art prosody transfer expressive TTS [58, 59, 60], which aimd at achieving controllable style synthesis in TTS. However, to learn and synthesize specific styles, there are limitations with unsupervised style factorization learning [60]. Since the disentanglement of different styles is heavily influenced by randomness and the choice of hyper-parameters [85], the learning of specific target styles is not completely controllable.

Under supervision with explicit prosody labels, the styles could be learned with direct guidance [39, 86]. Supervised learning requires a large amount of labeled data, resulting in difficulties for expressive TTS research and applications. Furthermore, the data labels for styles may not overlap well with our needs. An approach to tackle this is proposed in [76]. But, the external dataset and the synthesis dataset Blizzard 2017 [87] have differences in background noise, recording environment, speech quality, *etc.* With the differences between these two datasets, the classifier trained using an external dataset may not be well-adapted to extract representations from the synthesis data. The final emotion synthesis accuracy is 41% on four emotions [76] evaluated by listeners, which may be caused by the domain gap between the TTS dataset and the external dataset.

Datasets

TTS dataset

The TTS dataset was recorded in a voice production studio by multiple professional voice talents. The data are recorded with 24kHz sampling rate. It has balanced phonemic and textual information. After labelling appropriately for the task, 7% of the TTS dataset has utterance level style labels including happy, sad, neutral, angry, fast, and soft. Details of the data are summarized in Table 3.2. These utterances are used, as additional data, to train a multi-speaker style classifier, as described in the following section "Semi-supervised style transfer learning". To train the multi-style TTS, we use 40,244 utterances from a single speaker, including around 3000 style labelled utterances. The style embeddings for unlabelled portion are extracted using the style extraction model, more details of which are given in the following section "Semi-supervised style transfer learning".

IEMOCAP dataset

To compensate for the limited amount of labeled data in our TTS dataset, we chose IEMOCAP [82], which is widely used for emotion recognition, to complement our training data. In this dataset, both video and audio were recorded from ten actors in dyadic sessions under scripted and spontaneous communication scenarios. The dataset contains 12.5 hours of recordings with a sampling rate of 22kHz. Each utterance contains one emotion label, such as neutral, happy, sad, anger, surprise, *etc.* To be consistent with former research [76, 81] and also be suitable for our own interaction goal, we select the following emotions in our study: neutral, happy, sad, and angry. Similar to the approach in [81], we merge utterances with excited emotion with those of happy emotion.

Table 3.2: Training data style label statistics

Dataset	Split	Fast	Soft	Neutral	Happy	Angry	Sad
TTS Dataset	Train	1145	1814	4481	885	140	35
	Dev	105	161	439	79	13	3
	Test	124	220	506	93	17	2
	All	1374	2195	5426	1057	170	40
IEMOCAP	Train	–	–	1390	1307	865	883
	Dev	–	–	100	90	61	62
	Test	–	–	218	239	177	139
	All	–	–	1708	1636	1103	1084

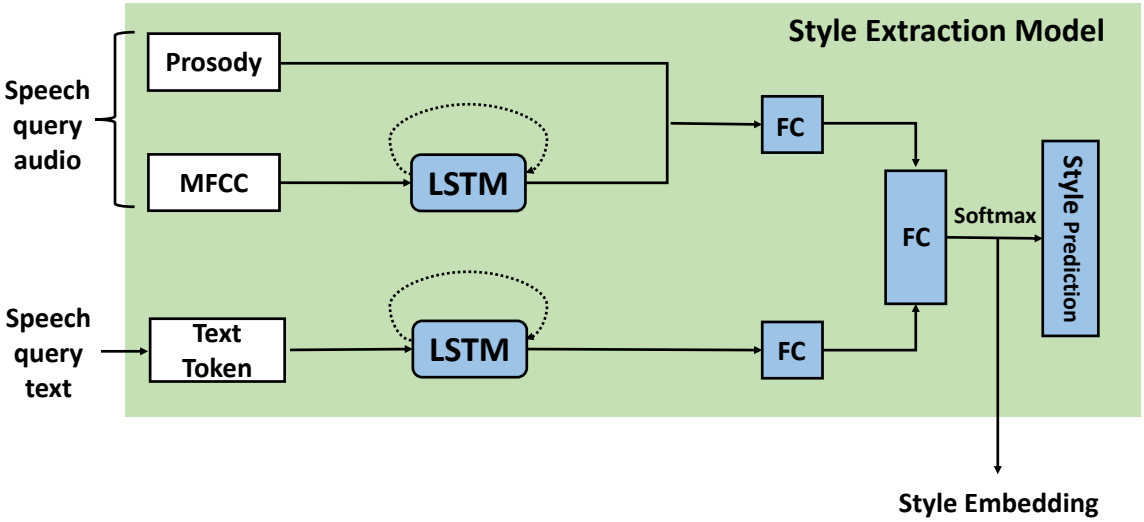


Figure 3-15: Multimodal style extraction model

Framework and Models

Semi-supervised style transfer learning

For speech style classification, we used the multimodal dual recurrent encoder (MDRE) model adapted from [81]. As shown in Figure 3-15, the model is composed of two separate recurrent encoders for audio and text modeling, respectively. The audio model uses 39-dimensional Mel-frequency Cepstral Coefficients (MFCC) features and utterance-level prosody feature extracted using openSMILE [88] as inputs, and the text model uses 300-dimensional embeddings to represent each word token. The MFCC, prosody, and text features are the same as those described in [81]. The audio

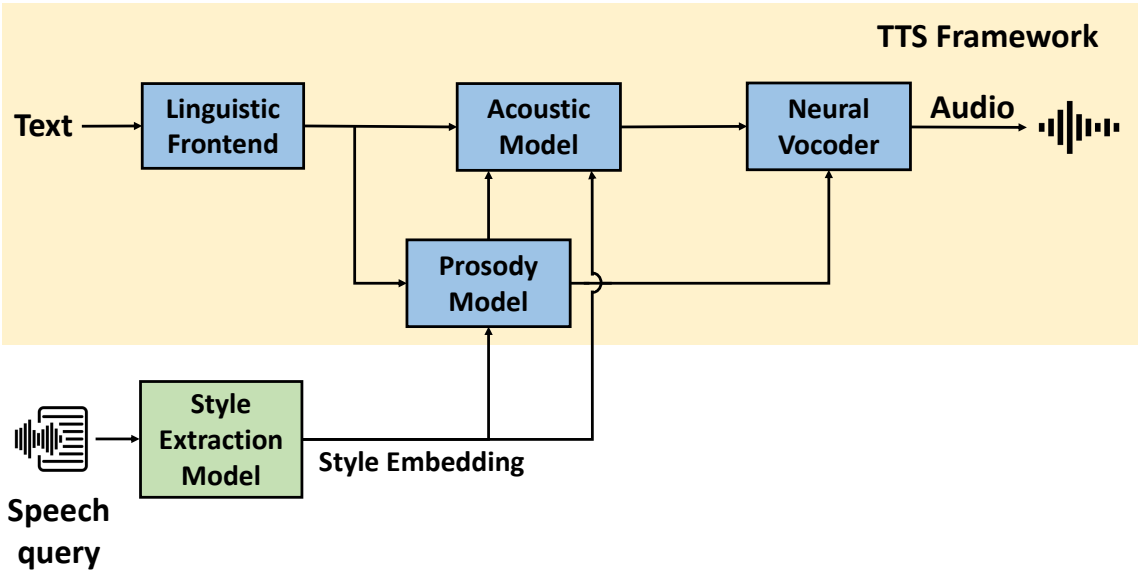


Figure 3-16: Style embedded TTS framework. The style extraction model generates the style embedding based on the user speech query, which is used to condition the TTS synthesis.

encoder output is concatenated with the text encoder output, then fed into a fully-connected layer to produce the final classification. We changed the loss function from sigmoid cross-entropy to softmax cross-entropy as it produced significantly better results for our training task. We use the softmax layer output as embedding features, which can be interpreted as a weighted representation of different speaking styles. The softmax feature as embedding is shown in Figure 3-15.

The style classifier is used to generate style embedding from the speech query during inference, as well as to extract style embedding for the TTS training dataset. At first, we trained the style classifier using the IEMOCAP dataset and applied it to generate style features on the TTS dataset. However, the classifier gives inaccurate predictions on the TTS dataset due to domain mismatch between the TTS dataset and the IEMOCAP dataset. Therefore, we labeled a small part of our TTS dataset and fine-tuned the style classifier using these labels.

Multi-style TTS system

Figure 3-16 shows the architecture of the expressive TTS system. It consists of a style embedding extraction component that generates the style embedding from speech query and a multi-style TTS, which uses the style embedding to synthesize its response in matching style. As shown in Figure 3-16, our TTS pipeline is a multi-model framework that consists of a linguistic frontend, a prosody model, an acoustic model, and a conditional neural vocoder. Specifically, the input text is first converted to linguistic features. Then, the linguistic features, along with any conditional features such as style embedding, speaker IDs are used to produce the prosodic features such as duration and F_0. The prosody model consists of a single layer LSTM model with 256 hidden units with content-based global attention [89], whose context vector contains linguistic features of the entire utterance. It is important to build a separate prosody model in the pipeline because it allows easier control for the speech style during synthesis time. Then, linguistic features combined with prosodic features are used to generate the 13-dim MFCC spectral acoustic features. The acoustic models consist of a two layer uni-directional LSTM with 256 hidden units per layer. At the last stage, a conditional neural vocoder using the WaveRNN [90], takes in the 13-dim MFCC along with the F0 feature to synthesize a 24kHz audio waveform. Our WaveRNN model consists of a single layer gated recurrent unit (GRU) with 1024 hidden units. The speaking style of the synthesized speech is controlled by the conditional style embedding feature, which can be pre-defined or extracted using the style extraction model from the input query, as in Figure 3-16.

Experiments and results

The style classification model is adapted from [81] and is shown in Figure 3-15. Specifically, we set the batch normalization layer with 0.9 momentum to help cross-domain adaptation. To compensate for the imbalance among style labels, we weighted the

Table 3.3: Style classification on TTS data. AdaBN helps the domain adaption between IEMOCAP dataset and TTS dataset, improving the weighted accuracy of six style classes.

Dataset	Trick	Neutral	Fast	Soft	Happy	Angry	Sad	**Accuracy**	
								Weighted	Unweighted
Train	BN	0.984	0.871	0.964	0.892	0.176	0.0	0.779	0.973
	AdaBN	0.953	0.847	0.918	0.903	0.353	0.0	0.915	0.957
Dev	BN	0.979	0.819	0.994	0.81	0.385	0.0	0.686	0.931
	AdaBN	0.927	0.8	0.963	0.873	0.538	0.333	0.766	0.904
Test	BN	0.984	0.871	0.964	0.892	0.176	0.0	0.683	0.940
	AdaBN	0.953	0.847	0.918	0.903	0.353	0.0	0.715	0.914

by-class loss function and the per-class accuracy with an inverse of style label prior and capped the neutral label prior to 0.25. Besides, AdaBN [91] is implemented in this model to boost domain adaptation performance between the TTS and multi-style datasets.

The multi-style TTS system is trained using the commissioned TTS dataset with style embedding features as conditional input features. The style embedding labels were generated by passing each utterance through the style classification model, as described in above. In the synthesis phase, the style embedding features could be automatically extracted from the input query or manually assigned as a combination of different styles.

In the style classification task, we first tested the style classifier model performance on the IEMOCAP train/test split. It achieves an overall accuracy of 72.7%, which is similar to the reported state-of-the-art [81]. To improve the embedding quality on the TTS dataset, the IEMOCAP dataset and the labeled subset of the TTS dataset were combined during training. The results show that the style classifier achieves 91.4% overall accuracy and 71.5% weighted accuracy on the TTS labeled dataset. With a lack of labeled data in anger and sadness in the TTS dataset, the prediction accuracy of these two classes is not high. The style classification accuracy decreased slightly

on the IEMOCAP dataset after joint training, likely due to the mismatch between the TTS and IEMOCAP datasets.

We performed normalization on the input features. The normalization is performed corpus-wise to compensate for the domain difference between our TTS dataset and the IEMOCAP dataset. Table 3.5 shows that normalizing both MFCC and prosody provides the best classification accuracy on the TTS dataset's validation set. Therefore, in the final model, we normalized both MFCC features and prosody features. The final classification accuracy for the TTS dataset is in Table 3.3.

Table 3.4: TTS data F0 statistics. The *Happy* style has higher mean F0 than other styles. The F0 variations of *Angry*, *Happy* and *Sad* are larger than *Neutral*, *Fast* and *Soft* styles.

Style	Angry	Happy	Sad	Neutral	Fast	Soft
F0	195.5±30.8	214.8±37.3	197.3±30.8	183.7±10.3	181.9±12.8	180.5±14.7

Multi-style TTS with conditional style embedding

To evaluate our multi-style TTS's performance, we collected subjective evaluation responses from 22 listeners. As reported in [76, 92, 93], the human perception on the emotions of natural speech is only around 50%, showing the ambiguity of emotion perception. Hence, instead of evaluating the subjective style accuracy on the multi-style synthesis results, we conducted the ABX test and preference test. Synthesis samples of our system are available at [94].

ABX test

The ABX test is designed to evaluate whether two styles generated with the same style embedding are perceived to be closer in speaking style when compared to a sample with a different style embedding. Since the style embedding can be used as a probability distribution over the 6 styles, to synthesize audio in a certain style, we

Table 3.5: Feature selection: Normalizing MFCC and prosody vector can improve the performance of style classifier.

Features	**Accuracy**	
	Weighted	Unweighted
Unnormalized	0.726	0.875
Normalized MFCC	0.673	0.840
Normalized prosody	0.494	0.62
Normalized both	0.766	0.904

construct the style embedding vector to have a value of 0.95 for the selected style and 0.01 for the other five styles. We designed the ABX test as follows. Given two different styles, we randomly choose an example in each style. We denote these two examples as A and B. We then randomly choose a different sample X from one of these two styles as reference. We then ask the listener to listen to samples A, B, and X, and then select which of A or B is perceived to be of the same style as the reference X.

We created 15 test sets in total, each of which corresponds to a pair of styles A and B, and a reference X. 22 listeners participated in the test, which gives a total of 330 ABX test comparison scores. We achieved an overall accuracy of 82.42% (*i.e.*, total number of matching pairs divided by the total number of ABX tests), indicating that the multi-style TTS is able to generate samples with perceivably distinguishable styles.

Preference test

The preference test is designed to compare TTS responses generated by a default TTS without multi-style capability and the multi-style TTS when the style embedding is explicitly provided. Specifically, we ask the listeners to choose between TTS responses synthesized with the same text but different models: baseline TTS

Table 3.6: Subjective preferences: The proposed TTS model's results are preferred over the baseline TTS model's.

	Baseline TTS	**Multi-style TTS**	
		Neutral Style	Other Styles
Preference (%)	28.0	54.2	17.8

model (*i.e.*, TTS without style embedding) or the multi-style TTS model. For the multi-style TTS responses, we provide either the neutral style or, when appropriate, a hand-crafted style embedding (*i.e.*, other style) assigned as a soft label whose style weights are determined based on the content of the utterance.

Results in Table 3.6 show that the multi-style TTS is preferred over the baseline TTS 72% of the time, indicating strong user-preference when an appropriately styled TTS response is provided. It is interesting to note that the neutral style from the multi-style TTS is preferred by the listeners most of the time. This is largely due to the content of the test utterances, which is best spoken with a peaceful and relaxing neutral style. This result is consistent with the findings in [76], which states that listeners prefer appropriate variation over random variation.

Mimicking real life input query with styled TTS response

We conducted experiments to evaluate the generalization capacity of the close-loop style extraction and multi-style TTS system. We recorded speech queries from multiple speakers who have never been seen in the training of our framework. These speakers read the queries freely in a quiet conference room. We then generated TTS responses for each query by conditioning on its style embedding. Our results show that over 40% of test pairs are evaluated as good matches by listeners. We noticed that when the speaking style of the input query is strong, the TTS response can match the input style to a certain extent (samples are at [94]). This can potentially be improved with more coherent style labels between the style extraction model training

data and the TTS dataset.

Discussions

In our proposed system, the soft style embedding is a weighted representation of different styles such that increasing the weight of a certain style emphasizes that style's effect on the synthesis outputs, shown in [94]. This demonstrates the multi-style TTS's capability of synthesizing styled-speech with respect to the soft style embedding. We noticed utterance-mean F0 differs for different styles in the synthesis results, representing the style difference. For example, the inference result of the Happy style has a significantly higher F0 mean than the other styles. This is consistent with the statistics of F0 for different predicted classes in TTS training data, as shown in Table 3.4.

In conclusion, we attempted to develop a style-embedded TTS that is more contextual and interactive. As shown in Section 3.2.2, with perfect style embedding, the system generated preferred TTS responses compared to a single style TTS. With automatically extracted style embeddings from real speech queries, the system demonstrated moderate capability in mimicking the speaking style of the input speech query. The overall quality can be improved with a more balanced multi-style TTS dataset and more coherent style labels between the style extraction model training data and the TTS dataset.

3.3 Key insights

As in the above TTS studies [95, 29], we also noticed that the multi-style TTS yielded poorer synthesis for the Happy style, which has a significantly higher F0 mean than the other styles. This could be due to the reason that the model focused on the most distinguishable feature, such as F0 mean, and failed to learn the nuances of the F0

contour. To mitigate this problem, the F0 mean and the F0 contour can be modeled separately. In addition, the sad and angry-styled audio quality was comparatively worse than other styles, which could be due to the lack of anger and sadness samples in the TTS dataset. In the future, the performance of the multi-style TTS system can be further improved with a training dataset that contains more balanced style labels.

Overall, for voice conversion (VC), the state-of-the-art VC is pushing edges on the any-to-any voice conversion using multi-speaker datasets and one-shot voice conversion that could generalize the voice conversion to unseen speakers. However, the utterances are usually short as two to four seconds, and there are noise, phoneme loss, as well as qualify decreasing in the converted voice according to the intrinsic disadvantages of voice conversion pipelines. You could hear some state-of-the-art samples from some recent papers' demo pages [96, 97]. Expression and style learning of the voice conversion is still a problem that needs future research to improve.

For text-to-speech, the models are usually trained using a large and high-quality dataset that leads to eventually better-generated speech quality compared to voice conversion. Multi-speakers TTS, prosody and styles embedded TTS are hot topics in the current research, and the quality of the expressiveness of generated utterances is improving. However, the shortcomings are that the TTS inference qualify is usually constrained by the training datasets. These datasets are usually high-quality reading datasets but may lack the style of real-life conversations. Therefore, the rare words or prosodies that are not well-covered by the training data will be challenging to capture in the TTS models.

Furthermore, we noticed that many natural aspects of human speech are not included in the current speech generation. For example, the filled pause is commonly existed in spontaneous speech, such as 'UM', 'UH'. These filled pauses have not been considered systematically in the current speech generation to have it con-

vincingly rendered in generated speech. Secondly, the unfilled pause, usually with background noise in real-life recordings, is usually replaced by pure silence in the speech generation, which is detectable and not natural. Thirdly, the breath sounds are not well-considered in the generated speech. In natural human speech, the inhalation/exhalation sound should be consistent. Other critical spontaneous speech phenomena such as repairs, repetitions, lengthenings, and discourse markers (*e.g.*, "like" and "you know") are also not well considered in the current mainstream TTS/VC research. Lastly, the natural prosodies (style, emotion, expressiveness, *etc.*) are still the cutting edge spots under development in machine learning based speech generation. Their quality in the synthesized/transformed speech still needs improvements.

We will further look at the voice transformation and text-to-speech artifacts in Chapter 4. With these shortcomings, we could develop features that capture the artifacts in the generated speech for better audio deepfake detection.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

Deepfake detection for practical systems

In this chapter, we will discuss the three approaches we take for robust deepfake detection. We start from features that capture signatures of human-generated speech, and then discuss the proposed features that capture signatures of machine-generated speech. Lastly, we will discuss the data-driven approach for deepfake detection and the issue of generalization.

4.1 Features that capture signatures of human-generated speech

The threats discussed earlier from the deepfakes indicate the urgent need for fake speech detection to assist the anti-spoofing capacity of ASV systems. To do so, our final goal is to find robust features for detecting fake speech, especially deepfakes. To re-iterate, our hypothesis is that features that capture the fine-level nuances of human speech from a speech-production perspective are likely to be able to effectively help distinguish between real and fake speech. In addition, they are also likely to improve

the performance of countermeasures that are used for thwarting ASV spoofing attacks carried out through synthetic speech.

4.1.1 Human voice-production-based features

In this section, the hypothesis is that machines cannot emulate many of the fine-level intricacies of the human speech production mechanism. We show that fundamental frequency sequence-related entropy, spectral envelope, and aperiodic parameters are promising candidates for robust detection of deepfaked speech generated by unknown methods.

Specifically, the hypothesis is that machine-generated speech is too consistent in many respects, and machines are unable to emulate the finer level variations found in naturally produced speech signals. In other words, because of the complexity of the human speech production mechanism, human speech has a greater degree of inconsistency than machine-generated speech. We devise experiments to investigate a select set of features that we believe capture some intricacies of human-generated speech in a manner that machines cannot.

In the production of speech, there are several sources that are either aperiodic or periodic that generate acoustic energy in the vocal tract. The aperiodic sources are aspiration generated at the glottis, friction generated in the vocal tract, and transient bursts from the rapid release of complete constrictions. The periodic source is the vibration of the vocal folds that creates periodic energy at the glottis. Identifying and quantifying these various sources has several applications in speech coding, speech recognition, and speaker recognition [98].

Synthetic utterances generated by deep generative systems lack specific aspects of naturalness. One notable example is that of prosody. While we do have high quality and plain prosody TTS datasets, these are far from perfect. This is likely to make prosody a promising candidate for our work. Prosody is partially represented

through variations in the fundamental frequency (F0) of the speech signal. In addition, **features that capture prosody variations** are the F0 sequence, spectral envelope, and spectral aperiodicity. We evaluate all of these in our work. Our hypothesis is that features that capture the fine-level nuances of human speech from a speech-production perspective are likely to be able to help distinguish between real and fake speech effectively. Besides, they are also likely to improve the performance of countermeasures that are used for thwarting ASV spoofing attacks carried out through synthetic speech. For example, as shown in Fig. 4-1, the spectral envelope information of fake speech lacks natural transition and nuances, consistent with our hypothesis that the synthetic utterances may lack some aspects of naturalness.

In the vocal production process of a human, the fundamental frequency we refer to is the natural frequency of the vibration of the vocal cords. A specific nuance we can leverage is (known from prior literature) that the larynx can be approximated a nonlinear dynamic system, and the vocal folds can be approximated to coupled oscillators that are theoretically capable of an infinite number of different vibration patterns. However, these are persistently in a perturbed state. In vocal acoustics, perturbation typically refers to a deviation from an expected regularity in vocal-fold vibration. No biological system can produce truly periodic oscillations, and some instantaneous fluctuation can always be expected [99]. **Features that capture such instant-to-instant perturbations** are the well-studied *jitter* and *shimmer* measurements, which gauge the cycle-to-cycle variations in frequency and amplitude of the speech signal, respectively. It is expected that the information captured by jitter and shimmer may be differently "enacted" in machine-synthesized speech (if at all). We thus choose also to evaluate these features in our work.

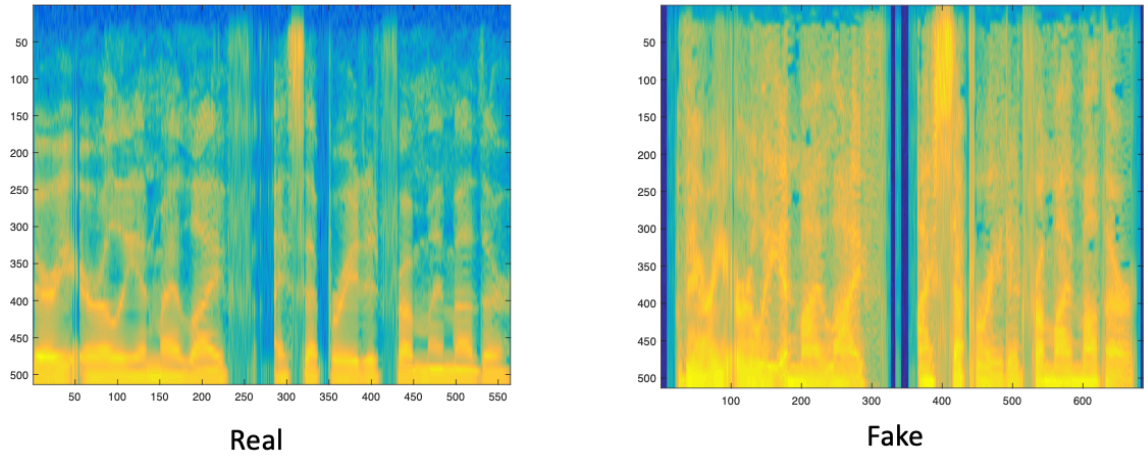


Figure 4-1: Spectral envelopes of real speech and fake speech. The same text utterance ("Very early in my life, I separated from my mother.")'s spectral envelope (log scale) of real speech contains natural transition and nuances while the fake speech does not. The short pause is unnaturally sharp in fake speech.

4.1.2 Analyzing the robustness of voice-production-based features

We are now in a position to analyze the robustness of speech production motivated features for detecting fake speech and improving the robustness of ASV systems against synthetic speech-based attacks. In fact an ASV countermeasure model that evaluates verification performance (based on t-DCF [100] and EER measurement) automatically consolidates and verifies both goals. For reasons explained earlier, we choose to use jitter, shimmer, and other features that capture F0 variations.

For experiments with jitter and shimmer, we only use the utterance-wise average jitter and shimmer values (extracted using the Praat [101, 102] python implementation [103]), which may not be the best way to use such transient information from speech signals. Nevertheless, we build a three-layer MLP as a countermeasure model that uses these features. In our implementation, we set the F0 range to be within 75-500 Hz. The results show a 31% EER on the development set, showing that even simple aggregates of these features (the average across an utterance in this case)

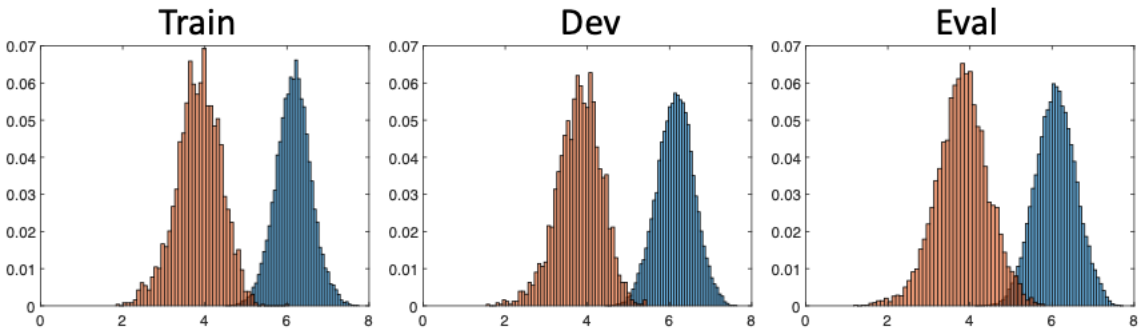
Table 4.1: ASV countermeasure-based evaluation

	Countermeasure EER%		t-DCF	
Features	DEV	EVAL	DEV	EVAL
Aperiodic parameters (AP)	21.19	20.65	0.4374	0.4445
Spectral envelope (SP)	10.55	9.31	0.3520	0.2453
MFCC	7.14	11.64	0.1942	0.2663
Spectrogram	0.48	9.39	0.0132	0.1954
AP+SP	9.41	8.91	0.2872	0.2462
AP+SP+MFCC	5.14	8.48	0.1560	0.2169
AP+SP+Spectrogram	0.62	6.67	0.0201	0.1604

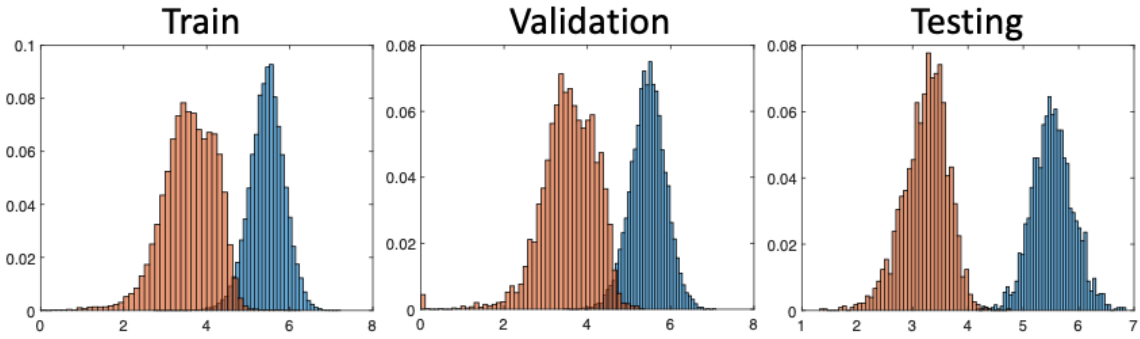
already make a positive difference to performance.

For experiments with aperiodic and spectral envelope signal features, we verify the spoofing countermeasures in performance improvements. We use the **detection model** that is modified from the residual net architectures proposed in [14]. To compare features, we do not focus on fine-tuning parameters and use five residual blocks compared to the 9-11 blocks in [14] for all input features. We set the kernel to be of different sizes to accommodate the dimensionality requirements of the spectral envelope and aperiodic information extracted using WORLD [104]. In our evaluation of these features, from Table 4.1, the EERs are similar for the dev and eval set using these features alone. We can also see that the fusion of aperiodic information and spectral envelope with mfcc or spectrogram can improve the detection performance as evaluated by EER and the joint performance with ASV evaluated by the t-DCF [100, 12] and decrease the gap between the EERs of the evaluation set and development set.

In the case of spectral entropy of F0 sequence, our hypothesis is that the F0 sequence of synthetic speech may lack the characteristic shift and variation of natural speech. We use the Shannon entropy of the power spectral density of the F0 sequence to capture this. The equations for the computation of this spectral density are as



(a) ASV dataset



(b) FoR dataset

Figure 4-2: Spectral entropy distributions. Blue is for fake speech and organge is for real speech

equation 4.1, equation 4.2 and equation 4.3, which first calculate the power spectral density (PSD) of the signal's spectrum $X(w_i)$, then normalize the PSD as probability density function, and finally compute the power spectral entropy.

$$P(w_i) = \frac{1}{N}|X(w_i)|^2 \tag{4.1}$$

$$P_i = \frac{P(w_i)}{\sum_i P(w_i)} \tag{4.2}$$

$$PSE = -\sum^{n} p_i \ln p_i \tag{4.3}$$

The F0 sequence is extracted using WORLD [104], and is trimmed to remove

the zero values at the beginning and end of the sequence. We plot the spectral entropy distributions for the ASVspoof 2019 logical data's train/dev/eval set and find consistent patterns in them. To evaluate the stability/significance of the patterns, we also compute the distribution from the FoR dataset, as shown in Fig. 4-2. Results show that the spectral entropy of F0 sequence is a surprisingly good indicator that captures statistical differences between synthetic speech and natural speech across datasets.

To further understand the anti-spoofing properties of the aperiodic signal and spectral envelope signals, we evaluated their performance through direct usage in the ASV model. As shown in Table 1.1 and Table 4.5, AP/SP-based black-boxes and white-boxes show much larger ASV EER% than STFT/MFCC based features under most attacks. This is even more obvious in SP-based boxes. The potential reason is that both AP and SP are features corresponding to identity-independent attributes like content-dependent attributes. SP is also mostly disentangled from speaker identities. These results are expected since the AP and SP signals are chosen to capture the nuances of differences between natural speech and fake speech, while ASV systems require features that distinguish the speakers' voice characteristics at a finer level. Still, one interesting phenomenon we noticed is that AP/SP features, especially AP, seem to be good as supplementary information that contribute to lower EER% for attacks, which STFT/MFCC are not good at.

4.1.3 Key insights

From the above study, we have established that features that capture the fine-level inconsistencies and nuances of the speech production process could consistently exhibit differences between synthetic speech and genuine speech. This is consistent with our hypothesis that they could capture the signature information to distinguish human-generated speech and machine-generated speech. The leverage of these hu-

man voice-production related feature could result in more robust detection of spoofed speech, and result in rendering ASV systems more robust to attacks generated using unseen methods.

4.2 Features that capture signatures of machine-generated speech

In this section, we will discuss the artifacts that are introduced by the generation methods of audio deepfakes. These artifacts can be used to develop audio deepfake detection methods.

There are two major approaches as discussed in previous sections for the generation of deepfake speech: text-to-speech and voice conversion. We will discuss them in the following subsections.

4.2.1 Artifacts in machine-generated speech

In the text-to-speech method, text is first processed to linguistic features, and then from linguistic features to acoustic features and finally from acoustic features to a waveform. During the process, there are many artifacts created [3]. Firstly, in the text to linguistic features step, there are inaccuracies in pronunciation prediction (*e.g.*, G2P), inaccuracies in text normalization as well as inaccuracies in prosodic feature prediction (*e.g.*, tone, pausing). Secondly, in the regression step of linguistic features to acoustic features, there are smoothing effects due to the statistical averaging and inaccuracies in statistical modeling for voicing prediction, pitch prediction *etc.* Lastly, during the waveform generation step, the predicted acoustic features will have artifacts from the prediction model such as the checkerboard effects of the deconvolutional neural networks [105]. Also the phase information is usually lost, and the remodeling of phase often creates discontinuities.

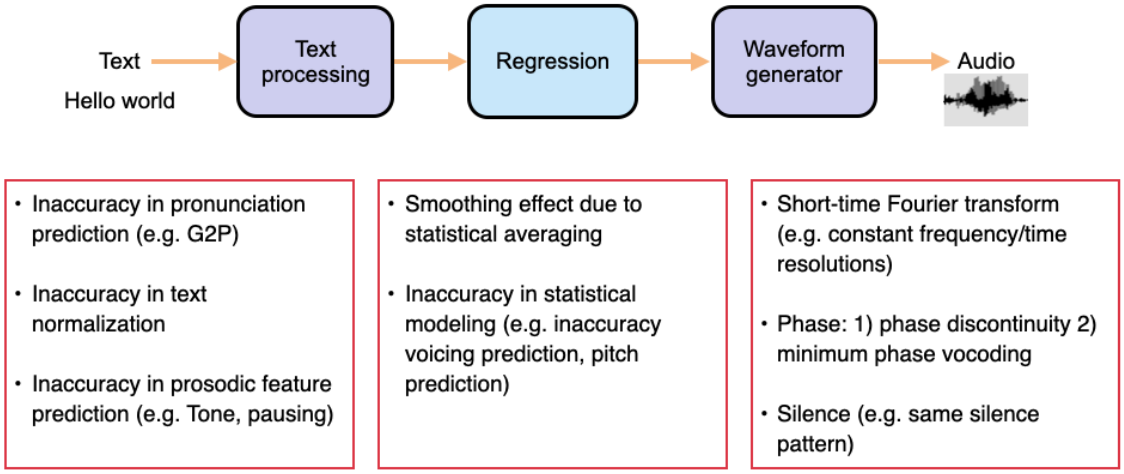


Figure 4-3: Artifacts in the text-to-speech [3]

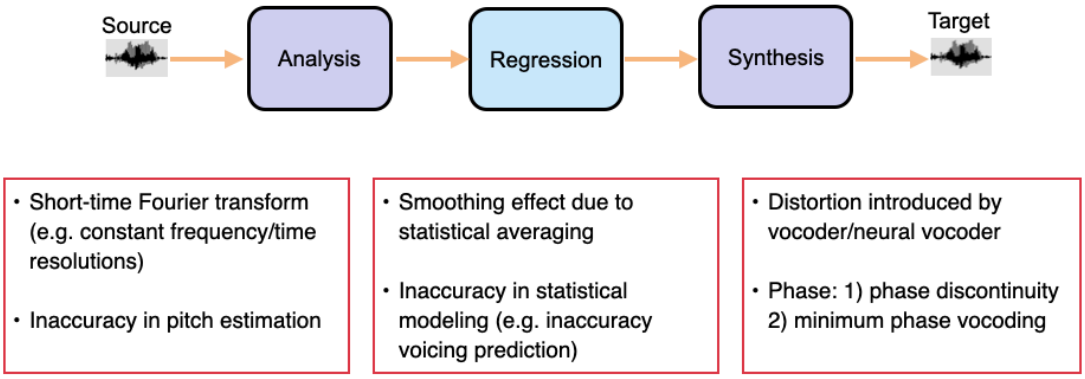


Figure 4-4: Artifacts in the voice conversion [3]

In voice conversion, the source utterance is transformed to the target utterance. There are artifacts from the pitch estimation, the smoothing effects of the learning model, as well as the distortions introduced by the vocoders. Phase discontinuity also exists.

4.2.2 Generalized spoofing detection inspired from audio generation artifacts

State-of-the-art methods for audio generation suffer from fingerprint artifacts and repeated inconsistencies across temporal and spectral domains. Such artifacts could be well captured by the frequency domain analysis over the spectrogram. Thus, we propose a novel use of long-range spectro-temporal modulation feature – 2D DCT over log-Mel spectrogram for audio deepfake detection. We show that this feature works better than log-Mel spectrogram, CQCC and MFCC as a suitable candidate to capture such artifacts. We employ spectrum augmentation and feature normalization to decrease overfitting to bridge the gap between training and test datasets with this novel feature. We developed a CNN-based baseline that achieved a 0.0849 t-DCF and outperformed the previously top single systems reported in the ASVspoof 2019 challenge. Finally, by combining our baseline with our proposed 2D DCT spectro-temporal feature, we decrease the t-DCF score down by 14% to 0.0737, making it a state-of-the-art system for spoofing detection. Furthermore, we evaluate our model using two external datasets, showing the proposed feature's generalization ability. We also provide analysis and ablation studies for our proposed feature and results.

Introduction

Audio deepfakes use deep learning and machine learning algorithms to generate or manipulate audio content with an intent to deceive. Such audio deepfakes are especially dangerous due to their innate embedding of biometrics, used in speech-based identity verification systems. State-of-the-art audio deepfake methods rely on voice conversion, text-to-speech synthesis, generative models, and neural vocoders [62, 59, 95, 106, 107]. With these advances, the quality of deepfakes has significantly improved, making them a pernicious means to commit a wide variety of fraudulent activities – identity theft and misinformation spread by untrained bad actors. Such

techniques even outperform professional human impersonators and threaten automatic speaker verification (ASV) systems [15].

For better spoof attack detection in ASV systems, ASV spoof challenges [16, 108, 17, 12, 109] have been created. In such challenges, the logical access (LA) consists of synthetically spoofed audio, which uses conventional signal processing and generative techniques that [110, 111, 112] propose the use of feature selection (*e.g.*, Constant Q cepstral coefficients [113], MFCC, log-Mel spectrogram, *etc.*), to search for the best features for spoof detection. However, these features have been developed for generic tasks, such as automatic speech recognition (ASR) and sound-based event detection, *etc.* They may not capture the fundamental differences between real and fake speech well. Further, the choice of feature selection can be influenced by audio datasets and is inconsistent. For better generalization, as noted in [15], unlike real speech, machine-generated speech consists of signature artifacts that can be leveraged for spoof detection. They propose a lightweight model with several human speech characteristics features and achieve comparably higher accuracy.

In computer vision, generative adversarial networks (GANs) [114] are a popular choice for image generation. Such methods have associated "fingerprint" [115] and signal-domain [116] artifacts that can be leveraged for detection and attribution studies. In speech synthesis, generative methods are used for feature learning from input linguistic features, while neural vocoders convert generated features into waveform outputs. Here, the audio is usually synthesized in frames or blocks of frames and has no cross-frame temporal consistency. This can lead to temporal modulation artifacts. Additionally, such methods are typically trained with element-wise mean-square-error losses in the Mel-Spectrogram domain [106, 117] and do not account for cross-frame consistency. Furthermore, speech is mainly encoded in the frequency ranges 0-4 kHz of auditory perception (based on the learning principles). There are associated artifacts with the generated outputs [118], especially at high frequencies [119]. The

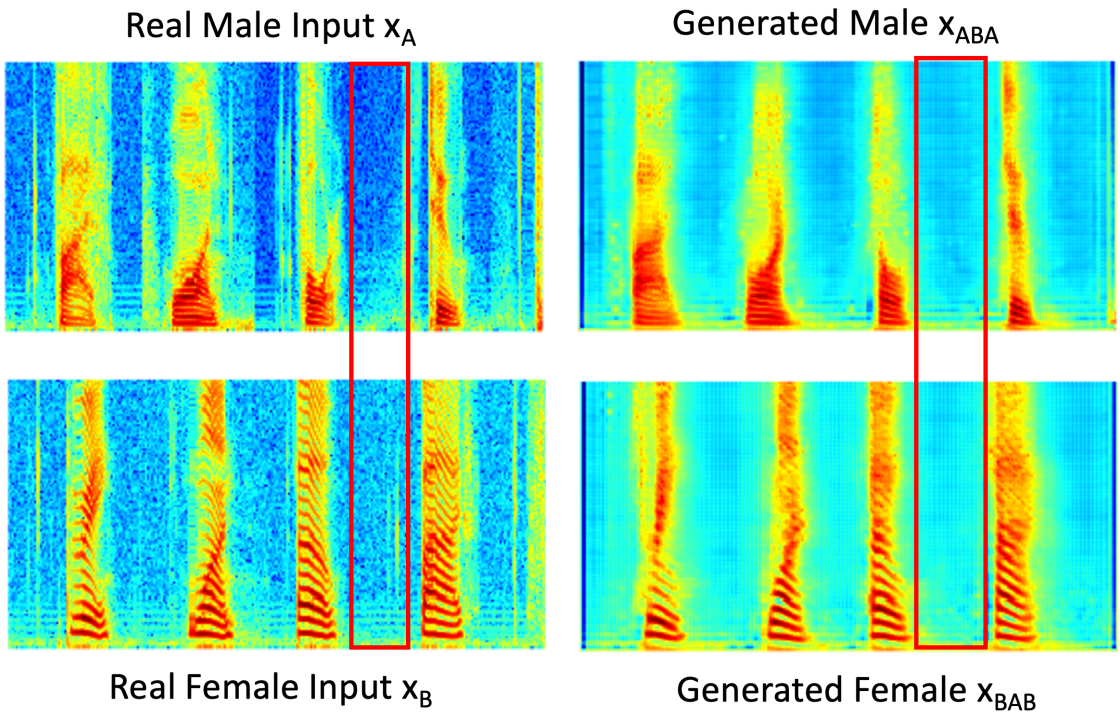


Figure 4-5: Checkboard effects in feature generation. These spectrograms are generated from our work of VoiceGAN [106]. The red boxes highlight the pattern differences.

Figure 4-5 shows an example of the feature artifacts in the generated features. These checkerboard effects may be preserved if the vocoders are signal processing based methods, such as Griffin-Lim method [53].

Based on these observations for feature artifacts, we propose using long-range frequency analysis on log-Mel Spectrogram (in feature domain) for spoof detection. Since 2D-DCT features capture repeated patterns/artifacts by analyzing the joint spectro-temporal modulation frequencies, we introduce the novel use of global 2D-DCT on log-Mel Spectrograms, a long-range spectro-temporal feature, to capture audio deepfake artifacts. The spoof detection convolutional neural network (CNN) classifier that operates on log-Mel Spectrum consists of the features with limited receptive fields and focuses on finding local short/medium time patterns/correlations in the input audio. The proposed global 2D-DCT feature essentially forces the CNN classifier to learn from the input audio's long-term/global modulation patterns. These

2D-DCT features correspond to the long-term spectro-temporal modulations rather than localized ones. Therefore, we call this proposed feature *global modulation* (Global M) feature. We show that the proposed feature detects deepfakes at a higher accuracy compared with the standard log-Mel features and could compensate our strongest baseline model to improve the overall detection performance further.

To summarize, in this section, we compare the proposed global modulation features with traditional features such as MFCC, log-Mel, and CQCC and present the following novel contributions:

1. We propose a novel long-range spectro-temporal feature – global modulation feature, for audio deepfake detection.

2. We further implement SpecAugment [120] and feature normalization to reduce over-fitting and bridge the gap between training and test dataset from unseen attacks.

3. The resulting baseline system achieves the best tandem detection cost function (t-DCF) scores as single systems according to [12]. Furthermore, our proposed feature can compensate for this strong baseline to bring the t-DCF and the equal error rate (EER) down and achieve state-of-the-art performance on the ASVspoof challenge 2019 logical access (LA).

Finally, the proposed global modulation feature also achieves a higher accuracy on general tasks, such as speaker verification, shown in Section 4.3.

Related works

Audio deepfake detection

The ASVspoof challenges [16, 17, 109] have raised efforts in fake speech spoofing attack countermeasures on ASV systems. Previous studies on anti-spoofing attacks on ASV systems and synthetic speech detection evaluate various features [112, 13] and

deep learning models [14] for detection performance. However, with the fast evolution of deepfake techniques, developing a detection system that is not constrained by the training data and can accurately detect new spoofed data generated from different or unseen deepfake algorithms is still challenging.

In the ASVspoof challenge 2019 dataset, the logical access (LA) component contains fake audio generated by multiple methods as in Table 4.6. As reported in [12], the best single system for LA data achieves a t-DCF metric [12] score of about 0.13 and an EER score of 5%. The top-3 primary system (a weighted voting of multiple systems) achieves a t-DCF score of less than 0.1 and an EER lower than 3%.

There are also datasets for audio deepfake detection like the FoR dataset [121] and the RTVCspoof dataset created using neural generation models as in [122]. In our work, we also use these external datasets effectively as unseen test attacks to our proposed detection system.

Modulation features

Modulation features capture the longer time patterns in the signal, which are often ignored in MSE-based generation [123, 117]. Motivated by generation artifacts, the proposed feature is a also global modulation feature that analyzes long-range spectro-temporal modulation.

In [124], the importance of the spectral and temporal modulation in the auditory spectrogram is discussed. Here, filter banks select different spectro-temporal modulation parameters ranging from slow to fast rates temporally and from narrow to broad scales spectrally. The spectro-temporal receptive fields (STRFs) of these filters are related to human perception. We also note that, from a physiological point of view, neurons in the primary auditory cortex of mammals are explicitly tuned to spectro-temporal patterns, *e.g.*, spectro-temporal features, [125]. Suthokumar *et al.* (2018) [126] analyze temporal modulation by performing FFT analysis in each sub-band, and show the effectiveness of temporal dynamics for detection of replay spoofing.

However, in previous studies, the 2D-DCT was only used to calculate **local** spectro-temporal modulation, such as for robust automatic speech recognition (ASR) [127]. Medium range modulation features were discussed in [128, 129] and long-range modulation was proposed in [130] – but both only for the temporal domain. Our **global** modulation feature combines spectral (as MFCC) and temporal modulation information for better long-range feature modeling. To the best of our knowledge, such long-range feature modeling has not been carried out in previous studies in speech.

Experiments

Baseline model

The baseline we use is a CNN-based model, similar to the baseline CNN model in [123]. As shown in Figure 4-6, the baseline model first consists of an initial convolutional layer followed by three residual blocks. Next, the output is passed through bidirectional Gated Recurrent Units (GRUs) and a self-attentive pooling layer. After temporal modeling and the self-attentive pooling, the feature vector is passed through a one-hidden-layer multi-layer perceptron (MLP) with two dimensions for the output. Finally, softmax is applied to obtain the prediction probability of genuine speech.

Proposed feature

The proposed feature is a simple and effective spectro-temporal feature: the 2D-DCT on log-Mel spectrograms. This is actually similar to the computation of Mel-frequency cepstral coefficients (MFCC) with the difference that we are applying a 2-dimensional (2D) discrete cosine transform (DCT) globally on both the temporal dimension and frequency dimension of the log-Mel spectrogram. The detailed computation steps are described as following:

a) Employ the fast Fourier transform (FFT) to compute the spectrum $X(w)$ of

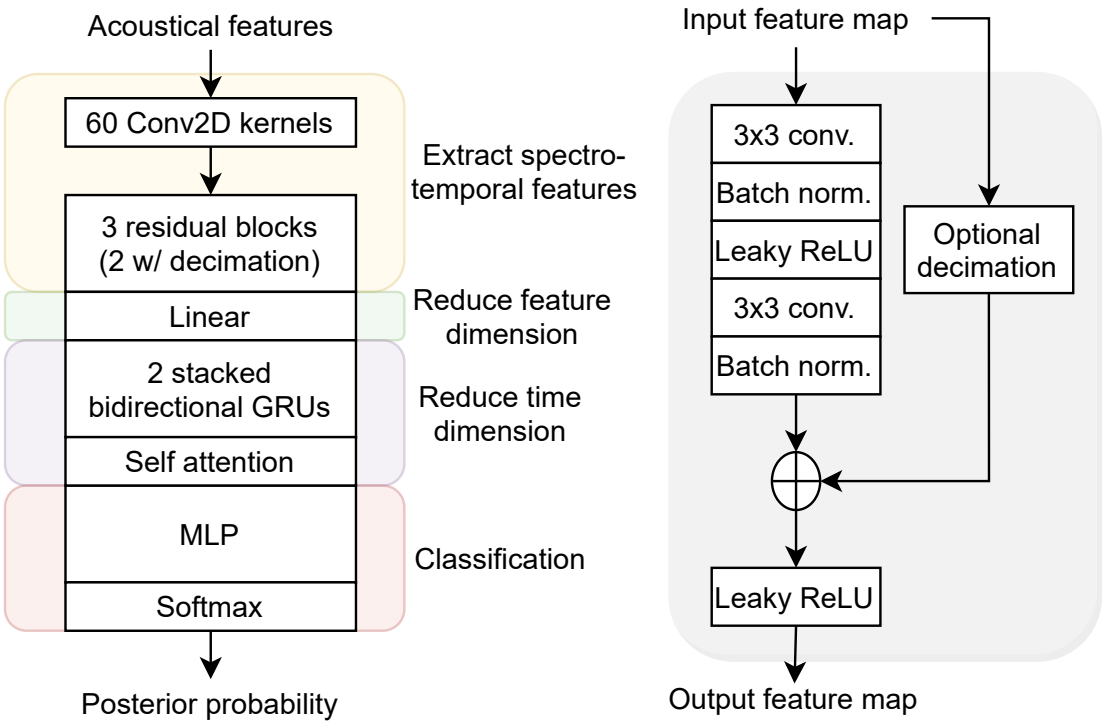


Figure 4-6: Block diagram of the baseline system (left) and the zoomed-in view of one residual block (right)

$x(n)$.

b) Compute power spectrum $|X(w)|^2$ and obtain the Mel-spectrum M by applying a Mel-frequency filter bank.

c) Apply multi-dimensional discrete cosine transform (DCT) to log-Mel to obtain dct_{n_M}.

d) Apply l_1-normalization or standardization normalization on the obtained dct_{n_M}.

Figure 4-7 shows the proposed 2D-DCT features for different spoofing types. The 2D-DCT features are shown in log-scale. From the visualization, we can see the proposed features exhibits differences in their patterns across different spoofing types. A17 and A19 use signal processing methods to generate fake audio, and the proposed features of these two are similar to bonafide audio. In contrast, other methods result in more complex changes compared to the bonafide (real audio) type.

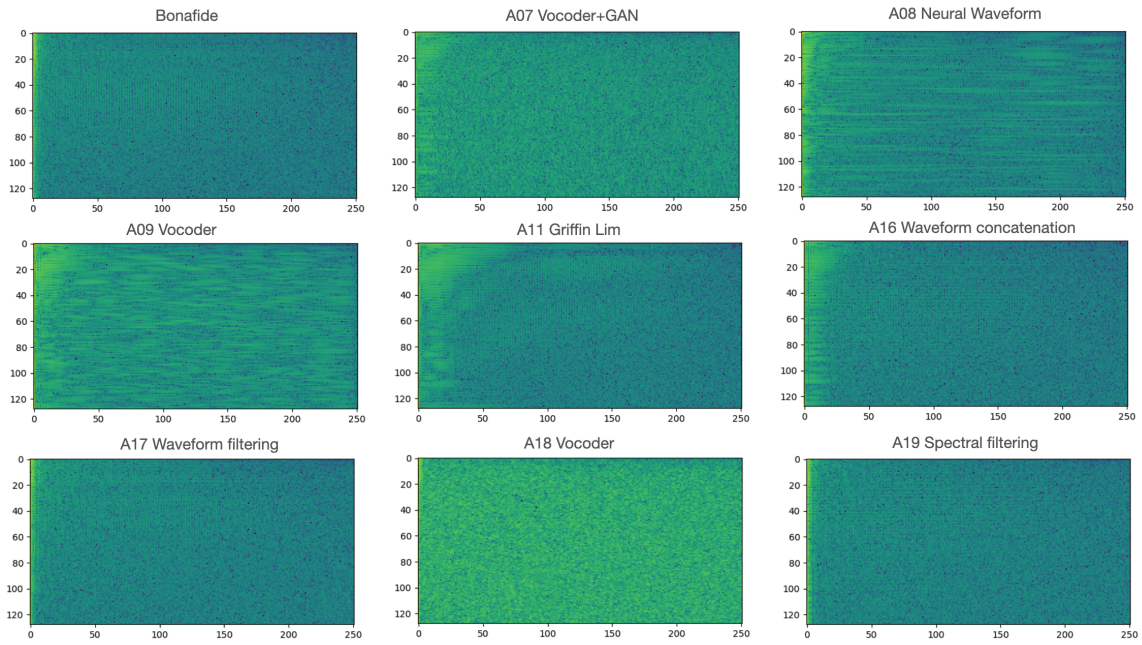


Figure 4-7: Visualization of the proposed features averaged within different spoofing types. Vertical axis is from mel-filters domain as spectro-modulation axis, and horizontal axis is from time frames as temporal-modulation axis. The proposed features exhibits differences in their patterns across different spoofing types (Best viewed zoomed in).

Implementation details

For experiments with conventional and proposed features, we verify spoofing countermeasures in terms of performance improvements. We use a detection model that is modified from the residual net architectures proposed in [14]. To evaluate the proposed features, we use a model similar to our baseline model without the attention layer since the temporal information is already condensed into the global DCT domain. The audio sequences are cut or padded to 4 seconds, as the temporal duration. The sampling rate is 16k, the FFT size is 1024, the window size is 512 and the hop size is 256, and the mel-filter number is 128. The details of the model implementation are in section 3.2 of [15]. Furthermore, we found the spectrum augmentation on the input features, and the normalization of the 2D-DCT features could improve the performance significantly, as shown in Table 4.2. We implemented the SpecAugment

Table 4.2: SpecAugment (SA) and normalization approaches

Features	**t-DCF**	**EER (%)**
log-Mel (Baseline$_1$)	0.0902	6.551
log-Mel w/ SA (Baseline$_2$)	**0.0849**	**5.139**
2D-DCT of log-Mel (Global M)	0.2851	12.40
Normalized Global M	0.1358	6.852
Normalized Global M w/ SA (Ours)	**0.1387**	**6.325**
T32 (Best single system [12])	0.1239	4.92

Table 4.3: Single system comparisons as ASV countermeasures

	Countermeasure EER%		t-DCF	
Features	DEV	EVAL	DEV	EVAL
Aperiodic parameters (AP)	21.19	20.65	0.4374	0.4445
Spectral envelope (SP)	10.55	9.31	0.3520	0.2453
MFCC	7.14	11.64	0.1942	0.2663
CQCC	1.37	10.89	0.0407	0.2746
log-Mel spectrogram	0.48	9.39	0.0132	0.1954
Normalized Global M	0.23	6.85	0.0067	0.1358
Normalized Global M w/ SA	0.17	6.32	0.0043	0.1387

(SA) [120] approach on log-Mel spectrograms with torchaudio. The random masking on the frequency channels and time steps of the spectrogram helps preventing overfitting and increases the model's performance [120]. For the SA on the proposed global modulation feature, a random zeroing-out is implemented to generate blank areas along both dimensions. This augmentation is only applied to the training data on the fly during training. Normalization of the 2D-DCT is performed using two approaches for comparison. The two normalization approaches, l_1-norm normalization and the mean/variance standardization, implemented using sklearn toolbox in Python, achieve similar results. In contrast, the normalization does not help much for the other traditional features since the values are already in reasonable ranges and the l_1-norm will break the spectral and temporal dynamics across the frames.

Results

Single systems and weighted voting scores

We evaluated the single system model taking in one type of feature and compared the proposed global modulation feature with the previously proposed "aperiodic signal" feature (AP), spectral envelope (SP) [15], and other conventional features such as MFCC, CQCC, and log-Mel spectrogram. To have a fair comparison, the model is the same ResNet model as in Section 3.2 of [15] with the last layer's dimension changed to facilitate the feature size difference. From the results in Table 4.3, we can see the proposed feature is significantly better in both the EER and the t-DCF scores than the other features. We further evaluate the joint performance of our proposed feature with the strong baseline models. We use different voting mechanisms for the joint scores between the Global Modulation feature and the baseline models as follows: For the prediction probability outputs of both systems, we weighted the prediction score using a ratio of 0.1 to 0.9. We use a max metric to keep the most confident voting among the two systems, which gives us the best performance. In contrast, the min-metric keeps the lower confidence prediction of the two joint systems. From the results in Table 4.4, we can see the joint scores improve the overall countermeasure performance.

Audio type analysis

To evaluate the detection performance on different spoofed-audio types, we do a comprehensive analysis of the t-DCF and EER scores for all spoofed-audio types in the LA evaluation set, as shown in Table 4.6. The A17 type, generated by waveform filtering manipulation of real audio, is visualized in Figure 4-7. It has a very similar modulation pattern to the bonafide audio and is the hardest type according to [109]. Our baselines and the proposed feature achieve top performance, compared to the EERs of single systems reported in [109]. Our joint system achieves one of the

Table 4.4: Weighted voting scores with different voting mechanisms

	Global Modulation + Baseline$_1$		Global Modulation + Baseline$_2$	
Ratios	t-DCF	EER	t-DCF	EER
min	0.1306	7.098	0.1230	6.636
0.0	0.1397	6.325	0.1387	6.325
0.1	0.1207	5.92	0.1253	5.778
0.2	0.1063	**5.89**	0.1141	5.780
0.3	0.0984	5.90	0.1057	5.631
0.4	0.0923	5.98	0.0994	5.520
0.5	0.0883	6.07	0.0930	5.542
0.6	0.0867	6.17	**0.0890**	**5.301**
0.7	**0.0865**	6.27	0.1057	5.563
0.8	0.0870	6.35	0.1142	5.778
0.9	0.0875	6.45	0.1253	5.929
1.0	0.0902	6.55	0.0849	5.139
max	**0.0737**	**4.03**	**0.0864**	**4.216**

Table 4.5: EERs of evaluation set for ASVspoof 2019 LA for speaker verification

	ASV EER%													
Spoofing ID	A07	A08	A09	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	ALL
STFT	2.33	2.65	3.75	47.56	40.89	47.59	37.01	29.09	35.48	4.09	12.07	28.61	1.88	22.24
MFCC	7.12	5.08	8.12	39.76	28.99	49.01	33.81	19.04	41.39	9.08	18.00	16.47	2.09	15.99
AP	38.93	32.46	32.59	42.37	38.29	43.28	37.02	33.96	41.12	49.06	40.05	34.57	44.53	39.25
SP	50.97	49.94	40.07	49.75	49.25	52.04	52.30	51.03	51.74	51.99	41.49	46.16	45.78	42.08
Global M	1.45	8.01	8.35	31.97	32.85	38.92	20.64	14.10	28.22	2.91	23.93	27.79	1.11	18.69

best three performances compared to all the other systems that use an ensemble of classifiers [12].

Speaker verification using the proposed features

To evaluate our proposed feature's effectiveness, we evaluate the feature under the automatic speaker verification scenario, as in [15]. The ASV model is trained with the ASVspoof 2019 data LA training set. We assign each spoofed utterance an identity that uniquely incorporates both speaker and attack. The 20 speakers and 6 types of attack in the ASVspoof2019 LA training set are combined into 120 "spoofed identities". With the bonafide audio, we have positive pairs, and negative pairs generated randomly in a balanced 1:1 ratio. The results are shown in Table

Table 4.6: Breakdown analysis of the performance on different spoofing audio types

Info			$Baseline_1$		$Baseline_2$		Proposed feature		Joint w/ $Baseline_1$		Joint w/ $Baseline_2$	
ID	System	Details	t-DCF	EER	t-DCF	EER	t-DCF	EER	t-DCF	EER	t-DCF	EER
A07	TTS	Vocoder+GAN	0.0000	0.0000	0.0000	0.0000	0.0054	0.1799	0.0014	0.0407	0.0020	0.0645
A08	TTS	Neural waveform	0.0463	1.4901	0.0163	0.5297	0.0521	1.9727	0.0147	0.5297	0.0254	0.7911
A09	TTS	Vocoder	0.0015	0.0577	0.0003	0.0170	0.0093	0.2852	0.0028	0.0815	0.0035	0.1392
A10	TTS	Neural waveform	0.0084	0.3022	0.0058	0.2445	0.0417	1.3208	0.0080	0.2852	0.0164	0.5059
A11	TTS	Griffin lim	0.0102	0.3667	0.0072	0.2852	0.0407	1.3038	0.0083	0.2682	0.0152	0.4720
A12	TTS	Neural waveform	0.0041	0.1222	0.0020	0.0645	0.0635	1.9557	0.0090	0.2852	0.0193	0.6111
A13	TTS-VC	WC + waveform filtering	0.0029	0.0985	0.0003	0.0170	0.0650	2.0372	0.0113	0.3429	0.0218	0.6689
A14	TTS-VC	Vocoder	0.0079	0.2445	0.0037	0.1222	0.0270	0.8149	0.0069	0.2274	0.0095	0.3022
A15	TTS-VC	Neural waveform	0.0186	0.5942	0.0061	0.1799	0.0248	0.7911	0.0069	0.2275	0.0097	0.3259
A16	TTS	Waveform concatenation (WC)	0.0007	0.0407	0.0005	0.0169	0.0062	0.1867	0.0010	0.0407	0.0016	0.0578
A17	VC	Waveform filtering	0.9760	44.486	0.7670	26.538	0.9017	36.286	0.8004	28.324	0.6218	28.405
A18	VC	Vocoder	0.0061	0.2037	0.0098	0.3259	0.1985	6.1286	0.0201	0.6111	0.0602	1.7927
A19	VC	Spectral filtering	0.0040	0.1222	0.0051	0.1630	0.0151	0.5297	0.0050	0.1799	0.0058	0.2037

4.5. The proposed feature is compared with results from other features given in [15]. Unlike AP and SP, the proposed 2D modulation feature is not only more powerful in a detection model but also effective in the audio type and speaker verification tasks. This clearly shows the potential of this proposed feature for several applications.

Discussions

As the above results show, our proposed global modulation feature has a strong performance compared to other conventional features. We also test our best model's detection accuracy on the other external datasets FoR [121] and RTVCspoof collected in [122]. For each dataset, 200 fake and 200 real samples are selected randomly from their test sets. Our Global modulation feature model could also predict the class of the randomly selected test data with reasonable accuracies of 90% to 98%.

We also compared the global modulation feature on the high-frequency section of the log-Mel spectrogram with the low-frequency section. Consistent with [119], the high-frequency section gives higher detection performance compared to the low-frequency section, although still not as good as using the global information altogether. Finally, we compare a blocked version of the modulation feature with our proposed global modulation feature. We did a simple 2×2 division on the log-Mel spectrogram and computed the 2D-DCT features separately for each block. The resulting localized modulation features give a significantly lower detection performance

of around 20% EER. This shows the importance of using long-range frequency computation to obtain the global inconsistencies for the audio-type detection. Interestingly, in [123], their proposed spectro-temporal receptive fields (STRFs) are a localized modulation feature. In their experiments for the ASVspoof challenge, they concluded that "the STRFs effectively reject distractor noise, but are by themselves not sufficient for discriminating real from synthetic speech". Their results, in comparison, give another evidence for the importance of computing the modulation features globally.

Also, it needs to be noted that the eval results for each feature are averaged across the eval EERs and t-DCFs from multiple runs for the soundness of the scores. The best eval score we have from a single running may be lower (*e.g.*, the best baseline we have has an EER of 4.03%). The t-DCF score is evaluated using the same metric as in [12].

Conclusions

In this section, we propose a simple yet effective feature, the global modulation feature, inspired by the fake audio's artifacts. We show that this proposed feature could improve the strongest baseline we have to further increase the countermeasure system's detection performance for the ASV system. Furthermore, we use this proposed feature to train our own ASV system and show that it also works very well for speaker verification tasks. This shows the broader potentials of the proposed global modulation feature.

In future works, we could adopt more data augmentation approaches, *e.g.*, adding noise, pre-processing with compression methods, *etc.* Moreover, with the future-released evaluation plan from ASVspoof challenge 2021, we would also evaluate the proposed feature's robustness to channel variations and its performance with the physical access (PA) dataset in ASVspoof Challenges [16, 108, 12, 109]. This work is published as in [131].

4.3 Data-driven approach for deepfake detection

The third approach is a direct data based detection. In this section, we will discuss more direct ways for audio deepfake detection focusing on data-driven learning and some related model improvements.

4.3.1 Advanced deep learning models and the issue of generalization

In this subsection, we will discuss the purely data-driven deepfake detection approach and some advanced deep-learning models. For data-driven deepfake detection, there are some related works that have been proposed for audio deepfake detection, such as [132, 133].

However, with the fast evolution of deepfake techniques, developing a detection system that is not constrained by the training data and can accurately detect new spoofed data generated from different or unseen deepfake algorithms is still a challenge. One key aspect of deepfake speech detection is to develop deepfake-speech detection algorithms that could be adapted to unseen deepfake speech generation methods. As in our paper [131], we proposed a strong baseline model that could have good performance in bridging the gap between validation set and evaluation set. The power of this model comes from the architecture design that use attention module and residual blocks.

More recently, in the ASVspoof 2021 challenge workshop, there were several papers proposing better models that could achieve higher spoofing detection performance. In [134], Tak *et al.* (2021) proposed a novel model named RawGAT-ST model to combine the spectral information with the temporal information at the model level. The proposed model extracts features from raw audio using a one-dimensional sinc convolutional layer and learns the relationship cues using a spectro-temporal graph

attention network (GAT). It achieves an equal error rate of 1.06% for the ASVspoof 2019 logical access database.

There are also interesting works using architecture search to find the best model architecture for speech deepfake and spoofing detection, such as [135]. In [136], Kang *et al.* (2021) compared several activation functions and analyzed their performance in the anti-spoofing systems. They also proposed an activation ensemble framework and compared the performance with single activation systems. And in [137], Chen *et al.* (2021) combined the Residual Neural Network architecture and different pooling techniques, and achieved very competitive results on the final evaluation set. They also investigated the effectiveness of the stochastic weight averaging and achieved competitive results in LA track.

Apart from model advancements, in [138], the self-supervised method is discussed following the PASE [139, 140] model that uses self-supervision to extract meaningful embeddings for a specific task. Furthermore, in the ASVspoof challenge 2021, data augmentation methods are encouraged to resolve the generalization problem. However, as in the challenge, all the proposed methods face significant performance drops in the evaluation set compared to their performances in the progress set. This gap suggests a high degree of overfitting on the progress set [141] and also indicates the generalization issue is still challenging and an open issue requiring further research.

4.3.2 Inductive learning for deepfake detection

As discussed above, one of the significant problems in audio deepfake detection is the threat from unseen samples, which may not have the same patterns or distribution as the training data. Adapting the model to new emerging attacks is a challenging problem. Ideally, we could continuously collect data as training data. However, it is expensive to collect and label them. Furthermore, collecting those samples will always be "late", since, by requirement, the threats must already be prevalent by the

time of collection.

In this subsection, we discuss a novel inductive learning approach to tackle this unseen data problem. We use inductive learning to add high confidence unseen attacks into the training data to adapt the pre-trained model. Different from transductive learning, which aims to label an unlabeled dataset based on the current model, inductive learning aims to not only label unlabeled datasets but also "produce a classifier" with the knowledge from unlabeled datasets. The unlabeled data could be used to adjust the pretrained classifier to better suit the unseen domain. This gives the model a way to employ those new data without labels.

In the ASVspoof 2021 Deepfake detection (DF) track, the data is comprised of bona fide and spoofed utterances generated using TTS and VC algorithms. The task is similar to the LA task (includes compressed data) but without speaker verification. In the DF task, general audio compressions (rather than telephony) is emphasized. In this year's challenge, no training data is provided and the labels of test data are not provided. We could only evaluate the performance of our system through an online evaluation platform. Therefore, the ASVspoof 2021 test phase could be transformed to a classical situation for inductive learning.

Since we need to use the online competition scoring system to evaluate the model performance, as the label information of test data is not published, we will derive labels for the test sets at first and then try to use these pseudo-labels to improve our system's performance. Basically, the induced pseudo-labels will be used to adapt the previous classifier, and the adapted classifier will then make new predictions for the entire test set, including the test samples with pseudo-labels that were previously added to the adaptation. This experiment is an interesting study to check the adaptability of the model under unseen attacks of deepfakes.

In the experiments, we use the baseline model in [123] and pass the test data through this baseline model. The initial evaluation performance is in Table 4.7.

Table 4.7: EERs of evaluation set for ASVspoof 2021 DF track

	DF-C1	DF-C2	DF-C3	DF-C4	DF-C5	DF-C6	DF-C7	DF-C8	DF-C9	Pooled
traditional vocoder	24.39	25.15	26.31	35.97	25.80	18.32	17.32	17.58	35.24	24.68
waveform concatenation	11.92	7.99	28.30	42.32	8.93	6.69	5.30	4.82	40.52	15.49
neural vocoder (autoregressive)	30.35	31.33	23.77	35.00	30.98	19.71	19.89	20.19	31.72	27.67
neural vocoder (non-autoregressive)	28.47	29.74	23.16	33.43	29.22	19.10	19.54	19.67	30.66	25.85
unknown	18.60	16.50	24.46	34.83	17.76	12.56	11.00	10.82	31.80	19.61
Pooled	26.64	27.41	24.99	35.09	27.66	18.62	18.32	18.45	33.32	25.41

Table 4.8: Performance of the inductive model for ASVspoof 2021 DF track

	DF-C1	DF-C2	DF-C3	DF-C4	DF-C5	DF-C6	DF-C7	DF-C8	DF-C9	Pooled
traditional vocoder	26.04	27.23	26.25	34.47	27.14	18.36	17.79	17.53	34.80	25.77
waveform concatenation	12.68	10.04	30.44	44.84	10.71	7.60	5.72	5.46	40.97	16.75
neural vocoder (autoregressive)	30.99	32.12	25.51	36.67	32.32	19.45	20.76	20.36	31.90	29.20
neural vocoder (non-autoregressive)	27.68	29.34	23.50	33.70	28.75	18.10	18.88	18.80	29.25	26.47
unknown	19.46	18.41	23.12	32.54	19.55	12.61	11.47	11.34	29.96	19.94
Pooled	27.51	28.73	25.56	34.81	28.66	18.44	18.66	18.48	32.60	26.56

We then select the predicted bona-fide and spoof audio with 1% top confidence as computed by our system and then use this section of data to retrain our baseline system for 5 epochs. Lastly, we use this adapted model to evaluate the testing dataset again. The predicted scores were submitted to the CodaLab system of this challenge [141]. The final performance is as in Table 4.8.

Different from our expectations, the results show that inductive learning does not help (EER dropped 1.15%) much in this situation. But this result is also understandable: with a baseline model that has converged and performs better on some families of synthetic speech, the top-confidence-score pseudo labeling may not help the model learn new information. The false positives will also pollute the re-learning. As a result, the model diverges to worse performance. To improve the performance, we could probably start from a model with better performance. Also, we could induce pseudo labels by different fake families separately, so the model could be re-trained in balance.

4.3.3 Data augmentation for deepfake detection

As we can see from the above subsections, a common challenge in the data-driven approach of audio deepfake detection, is the limitations in the data. The data size may not be large enough or varied enough to represent the situations in the test phase. One approach that might tackle this limitation and improve the detection performance is data augmentation, which has been proven to be an essential aspect for better detection.

For example, in the ASVspoof 2021 challenge, Das in [142] leveraged many data augmentation strategies to achieve good detection performance in the newest challenge. Furthermore, in [143], Tomilov *et al.* (2021) worked on all three tracks, and from the results, they found that several data augmentation methods work, especially with emulation of frequency distortions based on FIR filters. In [144], Chen *et al.* (2021) designed several data augmentation methods tailored to the ASVspoof 2021 challenge. They employed the ECAPA-TDNN [145] as the primary architecture and also adapted the channel-robust training strategies proposed in their previous paper [146]. These results show data augmentation in the right manner clearly helps to tackle the generalization issue of audio deepfake detection.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Conclusions

In this chapter, we will summarize the contributions of this thesis and also discuss some future directions that could be considered for generalized audio deepfake detection.

5.1 Summary and contributions

In this thesis, we started with a thorough experimental evaluation of the threats posed by audio deepfake, compared to other conventional spoofing methods. We showed the increasing threats from audio deepfake detection and its stronger attack ability compared to human impersonation and the 'shallow-faked' machine-generated speech. All these indicate the urgent need for audio deepfake speech detection, which could help ASV systems to be more robust to attacks generated using different methods.

We aimed to develop audio deepfake detection algorithms that could capture the fundamental differences between fake and genuine speech, *i.e.*, between machine-generated and human-generated speech. To do so, firstly, we traced back to the human speech production mechanism and showed that human speech has many special characteristics embedded into the voice. Especially, breath is an important feature of human speech, and inhalation sounds could contain biometric information about the

speaker.

We also studied state-of-the-art machine generation mechanisms to understand how audio deepfake are generated. We have done hands-on research on two mainstream generation methods: voice conversion and text-to-speech synthesis, focusing on impersonation of the target speaker's style. We observed the unnaturalness in the F0 pattern and the lack of modeling of spontaneous speech phenomena during the process.

Based on the previous studies on the generation mechanisms, we hypothesized that machine-generated speech produced by current technologies could not emulate many of the fine-level intricacies of the human speech production mechanism. Based on this hypothesis, we proposed features that are related to speech synthesis, which may be good candidates for audio deepfake detection. We showed that fundamental frequency sequence-related entropy, spectral envelope, and aperiodic parameters are promising candidates for the robust detection of deepfaked speech generated by unknown methods. Furthermore, features that capture instant-to-instant perturbations, such as jitter and shimmer, could also be used for audio deepfake detection.

Next, based on our understanding of the artifacts in text-to-speech and voice conversion, we proposed a novel long-range spectro-temporal feature – global modulation, for audio deepfake detection. This global modulation feature is derived from long-range 2D DCT applied to log-mel spectrograms, that combines spectral and temporal information. It is able to capture long-range dynamics and is sensitive to repeatable artifacts across frames and frequency ranges.

Lastly, we evaluated the data-driven approach for audio deepfake detection. Based on the limitation of the data sources and the fast advancing methods of audio deepfakes, generalization is an issue that needs future research to tackle.

We explored an inductive learning approach under the ASVspoof 2021 DF track. However, it did not improve our final model's performance. This is because of the

relatively big gap between training data and evaluation data. As a result, the baseline model converges to the samples in the training set, as a result of which the induced pseudo-labels contain many incorrect ones, especially false positive labels. In the future, a more generalized model could be a better starting point for this inductive learning approach.

In summary, the contributions of this thesis are:

- Firstly, we provided a thorough comparison of human speech and machine-generated speech and discussed the differences in their attack ability on ASV systems, to practically establish the importance of audio deepfake detection.

- Secondly, we studied the human speech production mechanisms and the machine speech generation mechanisms and gave insights to understand the fundamental differences between these two.

- Thirdly, we proposed several features that could capture the human speech production characteristics and machine-generated speech artifacts, which provide good performance in audio deepfake detection.

- Lastly, we also explored a data-driven approach and discussed the generalization limitations of the current research setting.

Our persepctive also motives many of the later works such as [147, 148]. We hope our contributions could inspire more future works in this direction. We also point out new data collections in the audio deepfake detection domain and increasing focus on this research area. We are glad to see emerging efforts in this area, such as the first Audio Deep synthesis Detection challenge [149].

5.2 Future directions and remaining challenges

With the emerging efforts in audio deepfake detection, there are several directions that are in development and have great potential. In the following, we will summarize three major directions that we believe are promising for the future research.

5.2.1 Data collection

There are still limitations with the current datasets. To have a better understanding how detection may face challenges from the new generation of deepfakes, a more comprehensive dataset is needed. There are several efforts in this field, such as [21, 149]. However, new generation algorithms trained using different languages and different speakers' data are appearing around the world on a daily basis. To win the battle with deepfake audio that is getting perceptually closer and closer to authentic speech, continuous and in-time collections of trending data are important. Ideally, automatic data-collection could be an exciting direction. Continuous data collection through web crawling using key words, active learning based on a strong base-model, or even manual collections and generations enforced by applications and company needs could be critical paths to a more comprehensive dataset of audio deepfakes.

5.2.2 Filled, unfilled pause and breath

As discussed previously in the dissertation, for the current TTS and VC techniques, utterance-level generation is the mainstream. For multiple sentences, filled or unfilled pauses inside speech are usually replaced using silence, which is an obvious target for detection. The breath sound is only naturally rendered inside the speech [32]. Moreover, the filled pause sounds (*e.g.*, 'uh', 'ah', *etc.*), are rarely considered in the generation of current deepfake speech [35, 33]. Algorithms that detect breath patterns and filled pause patterns can be leveraged for deepfake speech detection in long speech.

As an interesting finding in the ASVspoof 21 challenge [150], silence patterns in recordings may be an indicator to discriminate spoofs from real recordings. This is exactly one aspect of machine-generated speech since the silence has not been specifically modeled in the generation.

5.2.3 Segmental-level audio deepfake detection

Another trend in audio deepfakes is the segmental-level audio deepfake. This means the utterance is partially faked using a mixture of authentic speech and deepfaked speech.

In [151], a new dataset is proposed containing the partially fake audio. As in this paper, Yi *et al.* (2021) develop a dataset for half-truth audio detection (HAD). Partially-faked audio in the HAD dataset involves only changing a few words in an utterance. The audio of the words is generated with the very latest state-of-the-art speech synthesis technology. In [152, 153], Zhang *et al.* (2021) also deployed an initial investigation for detecting partially spoofed audio. They introduce a new database of partially-spoofed data, named PartialSpoof and investigate and compare the performance of countermeasures on both utterance- and segmental-level labels. Experimental results using the utterance-level labels reveal that the reliability of countermeasures trained to detect fully-spoofed data is found to degrade substantially when tested with partially-spoofed data, whereas training on partially-spoofed data performs reliably in the case of both fully- and partially-spoofed utterances. They concluded that spotting injected spoofed segments included in an utterance is a much more challenging task even if the latest countermeasure models are used. In ASVspoof 2021, Zhang *et al.* (2021) [154] followed their partialspoof work [153] and built multi-task learning (MTL) frameworks with squeeze-and-excitation (SE) blocks [155]. With their MTL framework, they tried to train one model for both utterance-level and segmental-level spoof detection and showed that the multitask learning improves the

detection performance for both tasks compared to their corresponding single training model. From the results, they also found that as expected, segmental detection is still more challenging than utterance-level detection.

From the above studies, we can see detecting segmental-level audio deepfake is still challenging and in development. More studies are on their way to support the reliable detection of partially-faked audio and to clear the threats from them using deepfake techniques.

Bibliography

[1] Charles A. Bouman. 0.13 lab 9a - speech processing (part 1). Purdue digital signal processing labs (ece 438), `https://www.jobilize.com/course/section/speech-production-lab-9a-speech-processing-part-1-by-openstax`, September 2009. Accessed: 2022–4-24.

[2] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning (ICML)*, pages 5180–5189. PMLR, 2018.

[3] Zhizheng Wu. Audio deepfake detection: An overview and its challenges. Seminar Lecture at John Hopkins University, `https://www.clsp.jhu.edu/events/zhizheng-wu-facebook-2/#.X6xI7ZNKjUL`, October 2020. Accessed: 2020–11-06.

[4] Lini T Lal and Avani Nath NJ. Identification of disguised voices using feature extraction and classification. *International Journal of Engineering Research and General Science*, 2015.

[5] Cuiling Zhang and Tiejun Tan. Voice disguise and automatic speaker recognition. *Forensic science international*, 175(2-3):118–122, 2008.

[6] Prabhu R Bevinamarad and MS Shirldonkar. Audio forgery detection techniques: Present and past review. In *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, pages 613–618. IEEE, 2020.

[7] Xiaodan Lin and Xiangui Kang. Supervised audio tampering detection using an autoregressive model. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2142–2146. IEEE, 2017.

[8] Robert Rodman and Michael Powell. Computer recognition of speakers who disguise their voice. In *The international conference on signal processing applications and technology ICSPAT2000*. Citeseer, 2000.

[9] Mireia Farrús. Voice disguise in automatic speaker recognition. *ACM Computing Surveys (CSUR)*, 51(4):68, 2018.

[10] Aäron 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. In *9th ISCA Speech Synthesis Workshop*, pages 125–125, 2017.

[11] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[12] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi H Kinnunen, and Kong Aik Lee. ASVspoof 2019: Future horizons in spoofed and fake audio detection. *Proc. Interspeech 2019*, pages 1008–1012, 2019.

[13] Md Sahidullah, Tomi Kinnunen, and Cemal Hanilçi. A comparison of features for synthetic speech detection. *Proc. Interspeech 2015*, pages 2087–2091, 2015.

[14] Moustafa Alzantot, Ziqi Wang, and Mani B Srivastava. Deep residual neural networks for audio spoofing detection. *Proc. Interspeech 2019*, pages 1078–1082, 2019.

[15] Yang Gao, Jiachen Lian, Bhiksha Raj, and Rita Singh. Detection and evaluation of human and machine generated speech in spoofing attacks on automatic speaker verification systems. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 544–551. IEEE, 2021.

[16] Zhizheng Wu, Tomi Kinnunen, Nicholas Evans, Junichi Yamagishi, Cemal Hanilçi, Md Sahidullah, and Aleksandr Sizov. ASVspoof 2015: the first automatic speaker verification spoofing and countermeasures challenge. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[17] Tomi Kinnunen, Md Sahidullah, Héctor Delgado, Massimiliano Todisco, Nicholas Evans, Junichi Yamagishi, and Kong Aik Lee. The ASVspoof 2017 challenge: Assessing the limits of replay spoofing attack detection. *Proc. Interspeech 2017*, pages 2–6, 2017.

[18] Rohan Kumar Das, Jichen Yang, and Haizhou Li. Long range acoustic and deep features perspective on ASVspoof 2019. In *IEEE Autom. Speech Recognit. Understanding Workshop*, 2019.

[19] Jichen Yang, Rohan Kumar Das, and Haizhou Li. Significance of subband features for synthetic speech detection. *IEEE Transactions on Information Forensics and Security*, 2019.

[20] Hossein Zeinali, Themos Stafylakis, Georgia Athanasopoulou, Johan Rohdin, Ioannis Gkinis, Lukáš Burget, and Jan Černockỳ. Detecting spoofing attacks using VGG and SincNet: BUT-Omilia submission to ASVspoof 2019 challenge. *Proc. Interspeech 2019*, pages 1073–1077, 2019.

[21] R. Reimao and V. Tzerpos. For: A dataset for synthetic speech detection. In *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 1–10, Oct 2019.

[22] Arsha Nagrani, Joon Son Chung, Weidi Xie, and Andrew Zisserman. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech Language*, 60:101027, 2020.

[23] Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee-Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han. In defence of metric learning for speaker recognition. *Proc. Interspeech 2020*, pages 2977–2981, 2020.

[24] Weicheng Cai, Jinkun Chen, and Ming Li. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system. In *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, pages 74–81, 2018.

[25] Tomi Kinnunen, Rosa González Hautamäki, Ville Vestman, and Md Sahidullah. Can we use speaker recognition technology to attack itself? enhancing mimicry attacks using automatic target speaker selection. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6146–6150. IEEE, 2019.

[26] Ville Vestman, Tomi Kinnunen, Rosa González Hautamäki, and Md Sahidullah. Voice mimicry attacks assisted by automatic speaker verification. *Computer Speech & Language*, 59:36–54, 2020.

[27] Xin Wang, Junichi Yamagishi, Massimiliano Todisco, Héctor Delgado, Andreas Nautsch, Nicholas Evans, Md Sahidullah, Ville Vestman, Tomi Kinnunen, Kong Aik Lee, et al. ASVspoof 2019: A large-scale public database of synthesized, converted and replayed speech. *Computer Speech & Language*, 64:101114, 2020.

[28] Ankit Shah, Hira Dhamyal, Yang Gao, Rita Singh, and Bhiksha Raj. On the pragmatism of using binary classifiers over data intensive neural network classifiers for detection of covid-19 from voice. *arXiv preprint arXiv:2204.04802*, 2022.

[29] Yang Gao, Weiyi Zheng, Zhaojun Yang, Thilo Köhler, Christian Fuegen, and Qing He. Interactive text-to-speech system via joint style analysis. *Proc. Interspeech 2020*, pages 4447–4451, 2020.

[30] Junichi Yamagishi, Koji Onishi, Takashi Masuko, and Takao Kobayashi. Modeling of various speaking styles and emotions for HMM-based speech synthesis. In *Eighth European Conference on Speech Communication and Technology*, 2003.

[31] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[32] Ulysses Bernardet, Sin-Hwa Kanq, Andrew Feng, Steve DiPaola, and Ari Shapiro. Speech breathing in virtual humans: An interactive model and empirical study. In *2019 IEEE Virtual Humans and Crowds for Immersive Environments (VHCIE)*, pages 1–9. IEEE, 2019.

[33] Rasmus Dall, Marcus Tomalin, and Mirjam Wester. Synthesising filled pauses: Representation and datamixing. In *9th ISCA Workshop on Speech Synthesis 2017*, pages 7–13, 2016.

[34] Eva Székely, Joseph Mendelson, and Joakim Gustafson. Synthesising uncertainty: The interplay of vocal effort and hesitation disfluencies. In *Proc. Interspeec 2017*, pages 804–808, 2017.

[35] Éva Székely, Gustav Eje Henter, Jonas Beskow, and Joakim Gustafson. How to train your fillers: uh and um in spontaneous speech synthesis. In *The 10th ISCA Speech Synthesis Workshop*, 2019.

[36] Wenbo Zhao, Yang Gao, and Rita Singh. Speaker identification from the sound of the human breath. *arXiv preprint arXiv:1712.00171*, 2017.

[37] Jonathan Fiscus, John S. Garofolo, Mark Przybocki, William Fisher, and David Pallett. 1997 english broadcast news speech (HUB4). In *LDC98S71*, USA, 1998. Linguistic Data Consortium.

[38] Tuomo Raitio, Ramya Rasipuram, and Dan Castellani. Controllable neural text-to-speech synthesis using intuitive prosodic features. *Proc. Interspeech 2020*, pages 4432–4436, 2020.

[39] Noé Tits, Fengna Wang, Kevin El Haddad, Vincent Pagel, and Thierry Dutoit. Visualization and interpretation of latent spaces for controlling expressive speech synthesis through audio analysis. *Proc. Interspeech 2019*, 2019.

[40] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

[41] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[42] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1857–1865, 2017.

[43] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[44] Ahmed Elgammal, Bingchen Liu, Mohamed Elhoseiny, and Marian Mazzone. CAN: Creative adversarial networks generating "Art" by learning about styles and deviating from style norms. In *8th International Conference on Computational Creativity, ICCC 2017*. Georgia Institute of Technology, 2017.

[45] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 700–708, 2017.

[46] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[47] Linguistic Data Consortium. TIDIGITS. `https://catalog.ldc.upenn.edu/LDC93S10`, 1993.

[48] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456, 2015.

[49] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[50] Yang Gao. Audio examples. `https://github.com/Yolanda-Gao/Spectrogram-GAN`.

[51] Wei Ming Liu, Keith A Jellyman, John SD Mason, and Nicholas WD Evans. Assessment of objective quality measures for speech intelligibility estimation. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE, 2006.

[52] Chanwoo Kim and Richard M Stern. Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis. In *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[53] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984.

[54] Yang Gao, Rita Singh, and Bhiksha Raj. Voice impersonation using generative adversarial networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2506–2510. IEEE, 2018.

[55] Ju-chieh Chou, Cheng-chieh Yeh, Hung-yi Lee, and Lin-shan Lee. Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations. *Proc. Interspeech 2018*, pages 501–505, 2018.

[56] Li-Wei Chen, Hung-Yi Lee, and Yu Tsao. Generative adversarial networks for unpaired voice transformation on impaired speech. *Proc. Interspeech 2019*, pages 719–723, 2019.

[57] Dongsuk Yook, In-Chul Yoo, and Seungho Yoo. Voice conversion using conditional cyclegan. In *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1460–1461. IEEE, 2018.

[58] Yuxuan Wang, RJ Skerry-Ryan, Ying Xiao, Daisy Stanton, Joel Shor, Eric Battenberg, Rob Clark, and Rif A Saurous. Uncovering latent style factors for expressive speech synthesis. *arXiv preprint arXiv:1711.00520*, 2017.

[59] RJ Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron Weiss, Rob Clark, and Rif A Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *International Conference on Machine Learning (ICML)*, pages 4693–4702. PMLR, 2018.

[60] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Fei Ren, Ye Jia, and Rif A Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *International Conference on Machine Learning (ICML)*, 2018.

[61] Daisy Stanton, Yuxuan Wang, and RJ Skerry-Ryan. Predicting expressive speaking style from text in end-to-end speech synthesis. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 595–602. IEEE, 2018.

[62] 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. *Proc. Interspeech 2017*, pages 4006–4010, 2017.

[63] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[64] Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan O Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. In *International Conference on Learning Representations (ICLR)*, 2018.

[65] Andrew Gibiansky, Sercan Arik, Gregory Diamos, John Miller, Kainan Peng, Wei Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural text-to-speech. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2962–2970, 2017.

[66] Sercan Ö Arik, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep voice: Real-time neural text-to-speech. In *Proceedings of the 34th International Conference on Machine Learning (ICML)- Volume 70*, pages 195–204. JMLR. org, 2017.

[67] Sercan Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice cloning with a few samples. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 10019–10029, 2018.

[68] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4480–4490, 2018.

[69] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5998–6008, 2017.

[70] 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.

[71] Simon King and Vasilis Karaiskos. The blizzard challenge 2011. In *Proc. Blizzard Challenge*, volume 2011, pages 1–10, 2011.

[72] Ryuichi Yamamoto. Pytorch implementation of convolutional networks-based text-to-speech synthesis models. github, `https://github.com/r9y9/deepvoice3_pytorch`, May 2018. Accessed: 2022–4-24.

[73] Yang Gao. Voice generation: Prosody transfer. github, `https://yolanda-gao.github.io/Prosodydemo/`, October 2018. Accessed: 2022–4-24.

[74] 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. *arXiv preprint arXiv:1609.03499*, 2016.

[75] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations (ICLR)*, 2015.

[76] Zack Hodari, Oliver Watts, Srikanth Ronanki, and Simon King. Learning interpretable control dimensions for speech synthesis by using external data. In *Proc. Interspeech 2018*, pages 32–36, 2018.

[77] Chi-Chun Lee, Emily Mower, Carlos Busso, Sungbok Lee, and Shrikanth Narayanan. Emotion recognition using a hierarchical binary decision tree approach. *Speech Communication*, 53(9-10):1162–1171, 2011.

[78] Emily Mower, Maja J Mataric, and Shrikanth Narayanan. A framework for automatic human emotion classification using emotion profiles. *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, 19(5):1057–1070, 2010.

[79] Kun Han, Dong Yu, and Ivan Tashev. Speech emotion recognition using deep neural network and extreme learning machine. In *Proc. Interspeech 2014*, 2014.

[80] George Trigeorgis, Fabien Ringeval, Raymond Brueckner, Erik Marchi, Mihalis A Nicolaou, Björn Schuller, and Stefanos Zafeiriou. Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5200–5204. IEEE, 2016.

[81] Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. Multimodal speech emotion recognition using audio and text. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 112–118. IEEE, 2018.

[82] Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335, 2008.

[83] Makoto Tachibana, Junichi Yamagishi, Koji Onishi, Takashi Masuko, and Takao Kobayashi. HMM-based speech synthesis with various speaking styles using model interpolation. In *International Conference on Speech Prosody*, 2004.

[84] Junichi Yamagishi, Makoto Tachibana, Takashi Masuko, and Takao Kobayashi. Speaking style adaptation using context clustering decision tree for HMM-based speech synthesis. In *2004 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages I–5. IEEE, 2004.

[85] Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *International Conference on Machine Learning (ICML)*, 2018.

[86] Gustav Eje Henter, Jaime Lorenzo-Trueba, Xin Wang, and Junichi Yamagishi. Principles for learning controllable TTS from annotated and latent variation. In *Proc. Interspeech 2017*, pages 3956–3960, 2017.

[87] Simon King and Vasilis Karaiskos. The Blizzard challenge 2017. In *Proc. Blizzard Challenge*, volume 2017, pages 1–10, 2017.

[88] Florian Eyben, Felix Weninger, Florian Gross, and Björn Schuller. Recent developments in openSMILE, the munich open-source multimedia feature extractor. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 835–838, 2013.

[89] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1412–1421, 2015.

[90] Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron Oord, Sander Dieleman, and Koray Kavukcuoglu. Efficient neural audio synthesis. In *International Conference on Machine Learning*, pages 2410–2419. PMLR, 2018.

[91] Yanghao Li, Naiyan Wang, Jianping Shi, Xiaodi Hou, and Jiaying Liu. Adaptive batch normalization for practical domain adaptation. *Pattern Recognition*, 80:109–117, 2018.

[92] Rainer Banse and Klaus R Scherer. Acoustic profiles in vocal emotion expression. *Journal of personality and social psychology*, 70(3):614, 1996.

[93] Klaus R Scherer, Rainer Banse, Harald G Wallbott, and Thomas Goldbeck. Vocal cues in emotion encoding and decoding. *Motivation and emotion*, 15(2):123–148, 1991.

[94] Yang Gao. Demo for "Interactive Text-to-Speech system via joint style analysis". `https://github.com/Yolanda-Gao/Interactive-Style-TTS`, 2019. Accessed: 2019-10-21.

[95] Yang Gao, Weiyi Zheng, Zhaojun Yang, Thilo Köhler, Christian Fuegen, and Qing He. Interactive Text-to-Speech system via joint style analysis. *Proc. Interspeech 2020*, pages 4447–4451, 2020.

[96] Ehab A. AlBadawy and Siwei Lyu. Voice conversion using speech-to-speech neuro-style transfer. In *Proc. Interspeech 2020*, pages 4726–4730, 2020.

[97] Zongyang Du, Berrak Sisman, Kun Zhou, and Haizhou Li. Expressive voice conversion: A joint framework for speaker identity and emotional style transfer. *arXiv preprint arXiv:2107.03748*, 2021.

[98] Om Deshmukh, Carol Y Espy-Wilson, Ariel Salomon, and Jawahar Singh. Use of temporal information: Detection of periodicity, aperiodicity, and pitch in speech. *IEEE Transactions on Speech and Audio Processing*, 13(5):776–786, 2005.

[99] James A Coan and John JB Allen. *Handbook of emotion elicitation and assessment*. Oxford university press, 2007.

[100] Tomi Kinnunen, Kong Aik Lee, Hector Delgado, Nicholas Evans, Massimiliano Todisco, Md Sahidullah, Junichi Yamagishi, and Douglas A Reynolds. t-DCF: a detection cost function for the tandem assessment of spoofing countermeasures and automatic speaker verification. In *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, pages 312–319, 2018.

[101] Paul Boersma and David Weenink. Praat: doing phonetics by computer [Computer program]. Version 6.1.38, retrieved 2 January 2021 `http://www.praat.org/`, 2021.

[102] Yannick Jadoul, Bill Thompson, and Bart de Boer. Introducing Parselmouth: A Python interface to Praat. *Journal of Phonetics*, 71:1–15, 2018.

[103] David Feinberg. Parselmouth praat scripts in python. Open Science Framework 10.17605/OSF.IO/6DWR3 `https://osf.io/6dwr3/`, 2019.

[104] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa. WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, 99(7):1877–1884, 2016.

[105] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.

[106] Yang Gao, Rita Singh, and Bhiksha Raj. Voice impersonation using generative adversarial networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2506–2510. IEEE, 2018.

[107] Takuhiro Kaneko and Hirokazu Kameoka. CyclegGAN-VC: Non-parallel voice conversion using cycle-consistent adversarial networks. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2100–2104. IEEE, 2018.

[108] Zhizheng Wu, Junichi Yamagishi, Tomi Kinnunen, Cemal Hanilçi, Mohammed Sahidullah, Aleksandr Sizov, Nicholas Evans, Massimiliano Todisco, and Héctor Delgado. ASVspoof: the automatic speaker verification spoofing and countermeasures challenge. *IEEE Journal of Selected Topics in Signal Processing*, 11(4):588–604, 2017.

[109] Andreas Nautsch, Xin Wang, Nicholas Evans, Tomi Kinnunen, Ville Vestman, Massimiliano Todisco, Héctor Delgado, Md Sahidullah, Junichi Yamagishi, and Kong Aik Lee. ASVspoof 2019: spoofing countermeasures for the detection of

synthesized, converted and replayed speech. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2021.

[110] Hong Yu, Zheng-Hua Tan, Yiming Zhang, Zhanyu Ma, and Jun Guo. DNN filter bank cepstral coefficients for spoofing detection. *IEEE Access*, 5:4779–4787, 2017.

[111] BT Balamurali, Kinwah Edward Lin, Simon Lui, Jer-Ming Chen, and Dorien Herremans. Toward robust audio spoofing detection: A detailed comparison of traditional and learned features. *IEEE Access*, 7:84229–84241, 2019.

[112] Madhu R Kamble, Hardik B Sailor, Hemant A Patil, and Haizhou Li. Advances in anti-spoofing: from the perspective of ASVspoof challenges. *APSIPA Transactions on Signal and Information Processing*, 9, 2020.

[113] Massimiliano Todisco, Héctor Delgado, and Nicholas Evans. Constant Q cepstral coefficients: A spoofing countermeasure for automatic speaker verification. *Computer Speech & Language*, 45:516–535, 2017.

[114] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems (NeurIPS) - Volume 2*, pages 2672–2680, 2014.

[115] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to GANs: Learning and analyzing GAN fingerprints. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7556–7566, 2019.

[116] Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *International Conference on Machine Learning (ICML)*, pages 3247–3258. PMLR, 2020.

[117] Tyler Vuong, Yangyang Xia, and Richard M Stern. A modulation-domain loss for neural-network-based real-time speech enhancement. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6643–6647. IEEE, 2021.

[118] Jordi Pons, Santiago Pascual, Giulio Cengarle, and Joan Serrà. Upsampling artifacts in neural audio synthesis. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3005–3009. IEEE, 2021.

[119] Xiaohai Tian, Zhizheng Wu, Xiong Xiao, Eng Siong Chng, and Haizhou Li. Spoofing detection from a feature representation perspective. In *2016 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 2119–2123. IEEE, 2016.

[120] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. SpecAugment: A simple data augmentation method for automatic speech recognition. *Proc. Interspeech 2019*, pages 2613–2617, 2019.

[121] Ricardo Reimao and Vassilios Tzerpos. FoR: A Dataset for synthetic speech detection. In *2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, pages 1–10. IEEE, 2019.

[122] Nishant Subramani and Delip Rao. Learning efficient representations for fake speech detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 5859–5866, 2020.

[123] Tyler Vuong, Yangyang Xia, and Richard M Stern. Learnable spectro-temporal receptive fields for robust voice type discrimination. *Proc. Interspeech 2020*, pages 1957–1961, 2020.

[124] Taishih Chi, Powen Ru, and Shihab A Shamma. Multiresolution spectrotemporal analysis of complex sounds. *The Journal of the Acoustical Society of America (JASA)*, 118(2):887–906, 2005.

[125] Marc René Schädler, Bernd T Meyer, and Birger Kollmeier. Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition. *The Journal of the Acoustical Society of America (JASA)*, 131(5):4134–4151, 2012.

[126] Gajan Suthokumar, Vidhyasaharan Sethu, Chamith Wijenayake, and Eliathamby Ambikairajah. Modulation dynamic features for the detection of replay attacks. In *Proc. Interspeech*, pages 691–695, 2018.

[127] Bernd T Meyer, Suman V Ravuri, Marc René Schädler, and Nelson Morgan. Comparing different flavors of spectro-temporal features for ASR. In *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[128] Hynek Hermansky and Nelson Morgan. RASTA processing of speech. *IEEE transactions on speech and audio processing*, 2(4):578–589, 1994.

[129] Hynek Hermansky and Sangita Sharma. Temporal patterns (TRAPS) in ASR of noisy speech. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 1, pages 289–292. IEEE, 1999.

[130] Zhizheng Wu, Xiong Xiao, Eng Siong Chng, and Haizhou Li. Synthetic speech detection using temporal modulation feature. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7234–7238. IEEE, 2013.

[131] Yang Gao, Tyler Vuong, Mahsa Elyasi, Gaurav Bharaj, and Rita Singh. Generalized spoofing detection inspired from audio generation artifacts. *Proc. Interspeech 2021*, pages 4184–4188, 2021.

[132] Tianxiang Chen, Avrosh Kumar, Parav Nagarsheth, Ganesh Sivaraman, and Elie Khoury. Generalization of audio deepfake detection. In *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, pages 132–137, 2020.

[133] Run Wang, Felix Juefei-Xu, Yihao Huang, Qing Guo, Xiaofei Xie, Lei Ma, and Yang Liu. Deepsonar: Towards effective and robust detection of AI-synthesized fake voices. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1207–1216, 2020.

[134] Hemlata Tak, Jeeweon Jung, Jose Patino, Madhu Kamble, Massimiliano Todisco, and Nicholas Evans. End-to-end spectro-temporal graph attention networks for speaker verification anti-spoofing and speech deepfake detection. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 1–8, 2021.

[135] Wanying Ge, Jose Patino, Massimiliano Todisco, and Nicholas Evans. Raw differentiable architecture search for speech deepfake and spoofing detection. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 22–28, 2021.

[136] Woo Hyun Kang, Jahangir Alam, and Abderrahim Fathan. Investigation on activation functions for robust end-to-end spoofing attack detection system. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 83–88, 2021.

[137] Tianxiang Chen, Elie Khoury, Kedar Phatak, and Ganesh Sivaraman. Pindrop labs' submission to the ASVspoof 2021 challenge. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 89–93, 2021.

[138] Ziyue Jiang, Hongcheng Zhu, Li Peng, Wenbing Ding, and Yanzhen Ren. Self-supervised spoofing audio detection scheme. In *Proc. Interspeech 2020*, pages 4223–4227, 2020.

[139] Mirco Ravanelli, Jianyuan Zhong, Santiago Pascual, Pawel Swietojanski, Joao Monteiro, Jan Trmal, and Yoshua Bengio. Multi-task self-supervised learning for robust speech recognition. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6989–6993. IEEE, 2020.

[140] Santiago Pascual, Mirco Ravanelli, Joan Serrà, Antonio Bonafonte, and Yoshua Bengio. Learning problem-agnostic speech representations from multiple self-supervised tasks. *Proc. Interspeech 2019*, pages 161–165, 2019.

[141] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, and Héctor Delgado. ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 47–54, 2021.

[142] Rohan Kumar Das. Known-unknown data augmentation strategies for detection of logical access, physical access and speech deepfake attacks: ASVspoof 2021. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 29–36, 2021.

[143] Anton Tomilov, Aleksei Svishchev, Marina Volkova, Artem Chirkovskiy, Alexander Kondratev, and Galina Lavrentyeva. STC antispoofing systems for the ASVspoof2021 challenge. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 61–67, 2021.

[144] Xinhui Chen, You Zhang, Ge Zhu, and Zhiyao Duan. UR channel-robust synthetic speech detection system for ASVspoof 2021. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 75–82, 2021.

[145] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification. *Proc. Interspeech 2020*, pages 3830–3834, 2020.

[146] You Zhang, Ge Zhu, Fei Jiang, and Zhiyao Duan. An empirical study on channel effects for synthetic voice spoofing countermeasure systems. *arXiv preprint arXiv:2104.01320*, 2021.

[147] Hira Dhamyal, Ayesha Ali, Ihsan Ayyub Qazi, and Agha Ali Raza. Fake audio detection in resource-constrained settings using microfeatures. *Proc. Interspeech 2021*, pages 4149–4153, 2021.

[148] Hira Dhamyal, Ayesha Ali, Ihsan Ayyub Qazi, and Agha Ali Raza. Using self attention dnns to discover phonemic features for audio deep fake detection. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1178–1184. IEEE, 2021.

[149] Jiangyan Yi, Ruibo Fu, Jianhua Tao, Shuai Nie, Haoxin Ma, Chenglong Wang, Tao Wang, Zhengkun Tian, Ye Bai, Cunhang Fan, et al. Add 2022: the first audio deep synthesis detection challenge. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 9216–9220. IEEE, 2022.

[150] Nicolas Müller, Franziska Dieckmann, Pavel Czempin, Roman Canals, Konstantin Böttinger, and Jennifer Williams. Speech is silver, silence is golden: What do ASVspoof-trained models really learn?

[151] Jiangyan Yi, Ye Bai, Jianhua Tao, Zhengkun Tian, Chenglong Wang, Tao Wang, and Ruibo Fu. Half-truth: A partially fake audio detection dataset. *arXiv preprint arXiv:2104.03617*, 2021.

[152] Lin Zhang, Xin Wang, Erica Cooper, and Junichi Yamagishi. Multi-task learning in utterance-level and segmental-level spoof detection. *arXiv preprint arXiv:2107.14132*, 2021.

[153] Lin Zhang, Xin Wang, Erica Cooper, Junichi Yamagishi, Jose Patino, and Nicholas Evans. An initial investigation for detecting partially spoofed audio. *arXiv preprint arXiv:2104.02518*, 2021.

[154] Lin Zhang, Xin Wang, Erica Cooper, and Junichi Yamagishi. Multi-task Learning in Utterance-level and Segmental-level Spoof Detection. In *Proc. 2021 Edition of the Automatic Speaker Verification and Spoofing Countermeasures Challenge*, pages 9–15, 2021.

[155] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 7132–7141, 2018.