Speaker Anonymization using End-to-End Zero-Shot Voice Conversion

by

Wonjune Kang

S.B., Massachusetts Institute of Technology (2020)

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of

Master of Science in Media Arts and Sciences

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author . . . .
Program in Media Arts and Sciences
May 13, 2022

Certified by . . . .
Deb Roy
Professor, Program in Media Arts and Sciences
Thesis Advisor

Accepted by . . . .
Tod Machover
Academic Head, Program in Media Arts and Sciences

Speaker Anonymization using End-to-End Zero-Shot Voice Conversion

by

Wonjune Kang

Submitted to the Program in Media Arts and Sciences,
School of Architecture and Planning,
on May 13, 2022, in partial fulfillment of the
requirements for the degree of
Master of Science in Media Arts and Sciences

Abstract

Spoken language is a rich medium of communication that combines words with various information about emotions, feelings, and excitation through modulations in tone and pitch. In discourse, this allows for maintaining a human element that is lacking in many other channels, such as writing or social media. However, a person's voice is a distinct biomarker, and there exist many settings in which it may need to be anonymized in order to protect the speaker's identity.

This thesis presents a framework for performing speaker anonymization using voice conversion (VC) methods. We first introduce a model for performing end-to-end zero-shot voice conversion by modifying the architecture of a neural vocoder. To the best of our knowledge, this is one of the first end-to-end approaches for zero-shot VC that has ever been proposed. Our model is able to maintain the clarity and intelligibility of transformed speech very well while also achieving good voice style transfer performance—an improvement over current state-of-the-art VC models, which exhibit a trade-off between audio quality and accurate voice style transfer.

Next, we present a method for extending targeted voice conversion to un-targeted voice anonymization. This is done by fitting a Gaussian mixture model (GMM) to the latent space of speaker embeddings that are fed into the VC model, and then sampling from the GMM to select the target voice for anonymization. This obviates the need for explicitly specifying a target speaker when performing VC-based anonymization.

We evaluate both our voice conversion and anonymization methods on publicly available data as well as real-world audio from conversations on the Local Voices Network (LVN) platform, demonstrating their applicability to "in-the-wild" settings. Finally, we provide a discussion of this work's potential applications and the ethical considerations of using voice conversion technologies in society.

Thesis Advisor: Deb Roy
Title: Professor, Program in Media Arts and Sciences

Speaker Anonymization using End-to-End Zero-Shot Voice Conversion

by

Wonjune Kang

This thesis has been reviewed and approved by the following committee members:

Deb Roy . . . .

Professor, Program in Media Arts and Sciences

Massachusetts Institute of Technology

Mark Hasegawa-Johnson . . . .

Professor, Department of Electrical and Computer Engineering

University of Illinois Urbana-Champaign

Kevin Esvelt . . . .

Associate Professor, Program in Media Arts and Sciences

Massachusetts Institute of Technology

Acknowledgments

During the last two years, I have had the opportunity to work with and learn from so many people, for which I am endlessly grateful. This thesis would not have been possible without them.

First, I would like to thank my advisor Deb Roy for guiding me and allowing me the freedom to explore different themes and topics in my research. During my first year, I had some difficulty in finding a research direction that resonated with me. I am truly appreciative of his support during this time, as well as for how he challenged me to think more critically about my research ethos and how I want to develop as a researcher. I would also like to thank my thesis committee members, Mark Hasegawa-Johnson and Kevin Esvelt, for their valuable feedback during the thesis writing process and for inspiring me as researchers.

A big thank you to all of the people in the Center for Constructive Communication, who have been fantastic mentors, collaborators, and friends. I would like to thank the recent alumni and graduate students who I have had the privilege of getting to know: Anneli Woolf, Nazmus Saquib, Juliana Nazaré, Nabeel Gillani, Prashanth Vijayaraghavan, Eric Chu, Will Brannon, Suyash Fulay, Maggie Hughes, Hang Jiang, Shayne Longpre, Bridgit Mendler, Cassandra Overney, Belén Saldías, Hope Schroeder, and Leonard Francis Vibbi. A special thank you goes to Doug Beeferman, Dimitra Dimitrakopoulou, Allen Gorin, and Brandon Roy, who have served as tremendous mentors to me in various capacities. And of course, thank you to Heather Pierce and Amy Johnson, who are the glue that keeps CCC together.

Thank you to my friends, both old and new, who were there for me and kept me sane in the heart of the COVID-19 pandemic: Haris Brkic, Chiho Im, John Kim, Henry La Soya, Justin Restivo, Luis Sandoval, Jocelyn Shen, Jenny Shi, Sriya Sreekanth, Evan Tey, Hanna Tseng, Bill Wu, and Jacob Zhang.

Most of all, thank you to my parents for their continual love and support, without whom I would never have been able to make it this far.

Contents

List of Figures

List of Tables

Part I

Overview

Chapter 1

Introduction

The past decade has seen the massive growth of speech technology-based services that many people use every day, such as personal voice assistants, Internet Protocol telephony, and social audio platforms. Many of these developments have been driven by rapid advances in algorithms for speech and audio-related tasks such as coding, recognition, and synthesis, which have in turn been spurred by advances in machine learning and the increased availability of massive datasets. The demand for these services is not surprising given the qualities of speech, which is a uniquely human mode of communication that conveys both linguistic and paralinguistic information such as emotion, emphasis, contrast, and focus [103]. Speech can also encode other elements of self-expression that may not be encoded by grammar or vocabulary, such as irony or sarcasm [30]. This makes it a much richer medium compared to other communication modalities such as written text.

However, there have been growing concerns in recent years about data privacy, especially in light of emerging problems with data ownership and violation of user trust by companies [31, 34]. This has led to increased demands for technologies that can preserve the privacy of various types of data. In the context of speech, which encodes a speaker's vocal identity, this raises interesting challenges for developing methods that can anonymize the identities of speakers.

Although there are several established methods for anonymizing speech, most of them, especially ones used commonly in media, result in unnatural, robotic voices that lose the expressivity and prosodic elements of the original speech. They can also result in the original utterances becoming garbled or less intelligible, which often means that subtitles must be used to allow listeners to properly understand what is being said. Moreover, some of these methods have the flaw of being reversible if the details of the original transformation are known.

One way of tackling the anonymization problem while maintaining naturalness and expressivity is through the lens of voice conversion (VC): the task of converting a speaker's voice to sound like that of another individual while maintaining the linguistic and prosodic content of the original speech [94]. Voice conversion can enable anonymization by transforming a speaker's voice to sound like that of someone else. In recent years, advances in deep learning have led to VC systems that can produce significantly more realistic and comprehensible converted voices compared to traditional methods. However, the naturalness, intelligibility, and accuracy of transformed voices even for state-of-the-art VC models still lags behind that of true speech, especially in settings where conversion is applied to new speakers that were previously unseen during model training (i.e., in the zero-shot setting) [86, 83, 124].

In this thesis, we are interested in the problem of anonymizing the voices of arbitrary speakers in multi-party conversations, where each speaker may have previously been unseen by the anonymization model and only a few minutes or seconds of speech data may be available per speaker. We approach this through the voice conversion paradigm, with the goal of synthesizing anonymized voices that sound natural and maintain the intelligibility, expressivity, and prosody of the original speech as much as possible. These conditions necessitate the development of a novel VC method that can transform voices with high fidelity in the zero-shot setting.

1.1 Motivation

In spoken language, much of the meaning is determined by context—that is, the objects or entities which surround a focal communicative event [103]. This means that the truth or validity of a proposition is determined by commonsense reference to experience. Consequently, spoken language tends to convey subjective information, and an important aspect of it is the establishment of a relationship between the speaker and the audience [103]. This contrasts with written language, in which there is a greater emphasis on logical and coherent argument, and most of the meaning is provided directly by the text itself.

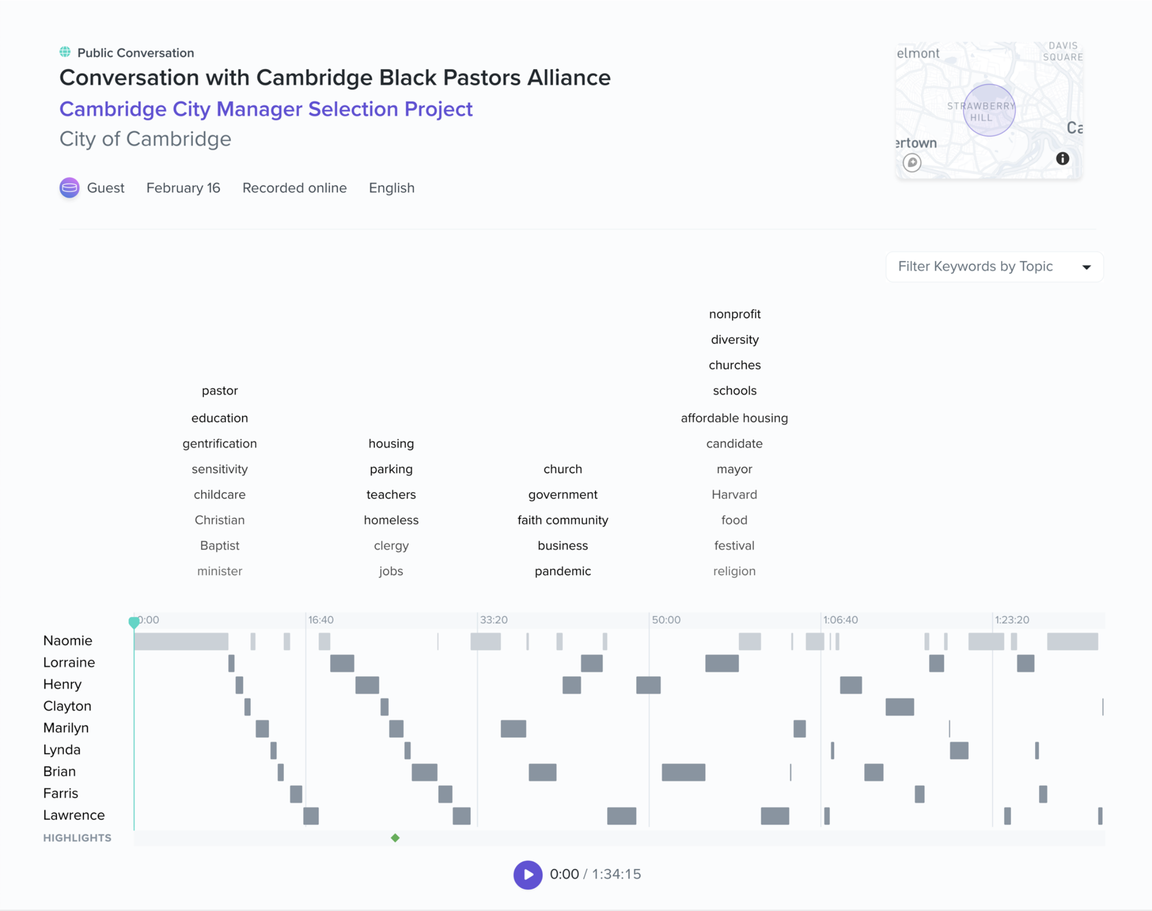
These properties of speech make it an ideal medium for people to share content such as stories and life experiences. Speech allows a listener to pick up on paralinguistic cues that convey the valence of emotional experiences [57, 89], and it has been shown to communicate human-like mental capacities related to thinking and feeling [90]. Consequently, listening to someone tell a narrative is often significantly more powerful and can convey its essence more effectively than reading a written version or transcript of the same story [58].

1.1.1 The Local Voices Network (LVN)

Motivated by these aspects of spoken dialogue, the Local Voices Network (LVN)[1] was established by the non-profit Cortico as a platform for hosting and collecting community-driven facilitated conversations on various social issues. LVN was created in order to combat the deterioration of social and institutional trust that has become prevalent in today's society [44], with the goal of building stronger civic spaces that can draw out and uplift conversation participants' life experiences. By creating channels where people can speak about social issues and listen to others, LVN conversations are meant to improve communication and understanding across various social divides. In doing so, they also aim to increase the opportunity for traditionally underheard communities to make their voices heard on important social issues. The platform has been successfully deployed in several real-world settings, including in Madison, Wisconsin in 2020 to elicit public opinion on the selection of a new police chief [59] and in Boston, Massachusetts in 2021 to draw out important public issues in the context of the Boston Mayoral Election [60].

Each conversation is led by a trained facilitator, who conducts the flow of the

[1] https://cortico.ai/platform/



Public Conversation

Conversation with Cambridge Black Pastors Alliance

Cambridge City Manager Selection Project

City of Cambridge

Guest February 16 Recorded online English

Figure 1-1: An example conversation recording on the online LVN platform that has been made available for public listening.

conversation and guides the other participants to share their experiences in relation to the topic at hand. Most conversations have between 4 to 8 participants and usually last between 40 to 90 minutes in length. All conversations are recorded, transcribed, diarized, and uploaded online; often, the conversation data is subsequently shared with others beyond the initial participant group for more public viewing. In this way, LVN enables the dissemination of peoples' voices in a fundamentally human way—by literally allowing others to listen to them speak. Figure 1-1 shows an example LVN conversation that has been made available for public viewing on the online platform.

Although LVN conversations are spaces in which participants are meant to be comfortable sharing their thoughts and experiences honestly, there are still settings in which they can be reluctant to do so for privacy reasons. These are often situations in which a participant may fear retaliation against them for what they say, such as if they share an opinion that would be perceived as deeply unpopular in their community or if they speak negatively of a party with the upper hand in a power dynamic. For this reason, many participants choose to use pseudonyms in LVN conversations. However, the fact that conversations are recorded and made available to others for listening means that participants may still be identified if someone recognizes their voice.

An anonymization system that could mask the identities of speakers in LVN conversations could have major implications for the content that is shared by participants, as they could be more willing to share certain stories if they had confidence that they could not be identified by others. To this end, the primary motivation of this thesis is to create a speaker anonymization system for speech in LVN conversations. In doing so, we must take care to preserve the emotion, expressiveness, and intelligibility of the original speech, which are aspects of spoken language that make LVN so powerful. Of course, such a system would also have applications that go beyond the LVN platform, as there are many other contexts in which expression-preserving voice anonymization can be valuable. These include online social audio platforms, anonymized interviews in media, and more generally, any setting in which a speaker would like to protect personal information such as geographical background or ethnicity.

1.2 Contributions

The key contributions of this thesis are as follows:

1. **A model for end-to-end zero-shot voice conversion.** We propose a novel approach for end-to-end zero-shot voice conversion that is based on the architecture of a neural vocoder. To the best of our knowledge, this is one of the first end-to-end zero-shot VC methods that has ever been proposed. Our model is able to strike a good balance between maintaining the clarity and intelligibility of transformed speech while also achieving good voice style transfer performance. This contrasts with current state-of-the-art VC models, which we find to exhibit a trade-off between audio quality and accurate voice style transfer.

2. **A method for extending targeted voice conversion to un-targeted voice anonymization.** We introduce a method that allows for the use of a voice conversion model to anonymize voices, even when a target speaker is not specified.

3. **Evaluation of the proposed voice anonymization framework on data from real-world spoken conversations.** We apply our voice conversion and anonymization methods to audio samples from conversations hosted and recorded on the Local Voices Network, demonstrating their effectiveness and applicability to "in the wild" audio that contains varying amounts of background noise and reverberation.

1.3 Outline

The remainder of this thesis is organized as follows:

- Chapter 2 discusses related work on the history and modern state of voice privacy preservation and voice conversion.

- Chapter 3 describes several key concepts and previous works that serve as foundations for our proposed voice conversion model.

- Chapter 4 describes the architecture and training procedure for our voice conversion model.

- Chapter 5 describes our method for extending targeted voice conversion to un-targeted voice anonymization.

- Chapters 6 and 7 present the data sources used for training and evaluation in this work and the objective and subjective metrics that are used for evaluation.

- Chapter 8 presents the results of using our model to perform voice conversion and anonymization, as well as performance comparisons against other current state-of-the-art VC models.

- Chapter 9 describes several experiments and explorations that were performed to better understand how our model works.

- Chapter 10 discusses limitations of the present work and directions for future research.

- Chapter 11 discusses the ethical considerations of using voice conversion technologies in society.

- Finally, Chapter 12 concludes with a summary of this work and final remarks.

Chapter 2

Background

In this chapter, we first present related work on voice privacy preservation in various settings and situate the present work within the greater literature. We then provide an overview of voice conversion as a task, discussing its theory, history, and some modern state-of-the-art methods.

2.1 Voice Privacy Preservation

With the rise of data-driven personalized systems in recent years [74], there has been growing demand for data privacy preservation around the world. Many countries and regions have enacted legal statutes to enforce the usage and sharing of data. One notable example is the European Union's General Data Protection Regulation (GDPR) [113], which regulates data protection principles when treating, transferring, or storing personal data. Hence, privacy-preserving data processing has become an active research area in many domains.

While a legal definition of privacy has yet to be established [66], speech contains a significant amount of personal information about the speaker that can be disclosed by various means [67]. For example, it is a distinct biomarker by which a speaker's identity can be determined by human listening or automated speaker recognition systems. In addition, a person's voice can reveal many facets of their identity, such as age, gender, ethnic origin, geographical background, and health or emotional state.

Consequently, there has been a great deal of interest in developing technologies for speech privacy preservation. These technologies can be used to mask speaker identities from human listeners [36], but increasingly, there have been efforts to develop systems that can do the same against automatic speech processing systems such as voice assistants [87, 2].

2.1.1 Methods

Modern approaches to speech privacy preservation can broadly be classified into four types: deletion, encryption, distributed learning, and anonymization [106]. Deletion methods [13, 21] are designed to obfuscate speech so that no information about it can be recovered; they are primarily meant for ambient sound analysis. Encryption methods [76, 6, 128] use cryptography to secure the raw data and support computation upon it in the encrypted domain. They tend to incur significant increases in computational load and can necessitate the use of special hardware. Decentralized or federated learning methods aim to train models from distributed data without accessing that data directly [51]. However, it has been shown that information about the original data in federated learning models can be derived by inverting gradients [18], which raises concerns about potential information leaks.

Speaker anonymization, which is addressed by this thesis, refers to the goal of suppressing the personally identifiable attributes of a speech signal while leaving most other attributes intact. Compared to the other privacy preservation methods described above, anonymization methods tend to be more flexible because they can selectively suppress or keep unchanged certain speech attributes and be easily integrated into existing speech processing systems. Previous approaches for anonymization have included noise addition [25], speech transformation [82, 77], voice conversion [80, 16, 24, 96], and disentangled representation learning [95, 1].

2.1.2 VoicePrivacy initiative

Recent increased interest in developing solutions for speech privacy preservation motivated the creation of the VoicePrivacy initiative [106, 107], an enterprise that aims to foster the development of technologies in this domain. The initiative aims to bring together a research community in order to formulate specific tasks of interest, develop evaluation methodologies, and benchmark new solutions through a series of challenges. Specifically, its mission is to "foster progress in the development of anonymization and pseudonymization solutions which suppress personally identifiable information contained within recordings of speech while preserving linguistic content, paralinguistic attributes, intelligibility and naturalness" [108].

The first VoicePrivacy Challenge [109] was organized in 2020 as part of this initiative, focusing on the task of voice anonymization, and laid out a set of common datasets, protocols, and metrics for evaluating anonymization performance of models. The second iteration of the VoicePrivacy Challenge is to happen later in 2022 [108]. The challenge's evaluation criteria place large importance on maintaining aspects of speech that are related to human perception, such as naturalness and intelligibility, which are evaluated using both objective and subjective metrics. Given the similarities between these criteria and our own, this work largely grounds its evaluation framework in the methodologies used in the challenge.

2.2 Voice Conversion

It is relatively simple to anonymize speech by modulating voice characteristics using signal processing techniques. Some basic methods involve altering the frequency spectrum to change the perceived pitch of the voice or designing acoustic filters to change the spectral characteristics of the speech. Many speech anonymization methods today, especially ones popularly used in the media, fall under these categories. However, these methods often result in robotic, unnatural transformed voices. Because they are not explicitly designed to retain the comprehensibility of the speech, they can also necessitate the use of subtitles or captions to allow listeners to understand what is being said. Moreover, some of these methods can be reversed if the details of the original transformation are known.

Therefore, this work approaches speaker anonymization through the perspective of voice conversion (VC). Voice conversion is the task of converting one's voice to sound like that of another person without changing the linguistic or prosodic content conveyed in the original speech [94]. It belongs to the general field of speech synthesis, which also includes text-to-speech (TTS), speech vocoding, and the changing of other speech properties such as emotion and accents. In addition to speaker anonymization, VC technology has many other real-life use cases, such as personalized speech synthesis, communication aids for the speech-impaired, and voice dubbing in movies.

2.2.1 Information in speech

A given speaker can be characterized by three high-level factors [94]: 1) linguistic factors, which are reflected in sentence structure, lexical choice, and idiolect, 2) suprasegmental factors, such as the prosodic characteristics of a speech signal, and 3) segmental factors related to short-term features, such as the spectrum and formants. When linguistic content is held constant, the suprasegmental and segmental factors are the relevant factors that encode information about a speaker's identity. Thus, a voice conversion system is expected to convert the suprasegmental factors and segmental factors of a source speaker to those of a target speaker.

In this work, we take a slightly coarser view of these factors. We refer to the linguistic information conveyed by a speech utterance as *content* information. The suprasegmental and segmental factors are grouped together under *pitch* (the fundamental frequency and rise or fall the pitch contour across syllables), *rhythm* (the speed at which a speaker utters combinations of phonemes), and *timbre* (the voice characteristics of a speaker that are reflected in formants encapsulated in the frequency components of the spectral envelope). While there has been some research on disentangling these various components from one another [84, 85], most VC systems today are only able to disentangle and change the timbre of a speech utterance. Therefore, we broadly denote the timbre as *speaker information* and frame the role of a VC system as converting the timbre of an utterance to sound like that of another target individual while maintaining the content, pitch, and rhythm.

2.2.2 Typical pipeline

A typical VC system takes as input a source utterance (from the original speaker) and a target utterance (from the intended new speaker). The objective is to extract the content information from the source utterance and combine it with the speaker information from the target utterance, resulting in a speech signal that corresponds to the content of the source utterance spoken in the target speaker's voice. A VC system usually includes three key modules: speech analysis, mapping, and reconstruction [94]. Figure 2-1 shows a block diagram of the overall pipeline.

The goal of the speech analysis module is to decompose the source utterance signal into some intermediate representation for effective manipulation or modification with respect to the acoustic properties of speech. These intermediate representations often take the form of log magnitude spectrograms taken from the short-time Fourier transform (STFT), sometimes mapped onto the mel scale.

The core of a VC system is the mapping module, which performs the actual speaker conversion function. This module performs a transformation on the intermediate representation from the speech analysis module, combining the speaker information from the target utterance with the content information of the source utterance. For example, if the input intermediate representation is a log-mel spectrogram, the mapping module outputs another log-mel spectrogram. The new spectrogram should theoretically correspond to a time domain speech signal where the source utterance is being spoken in the target speaker's voice.

Finally, the reconstruction module converts the transformed intermediate features back into a time domain speech signal. This module can be implemented using phase reconstruction methods such as the Griffin-Lim algorithm [23], but modern approaches usually use neural network-based vocoders [73, 81, 47, 35], which are able to reconstruct speech with much higher fidelity.

Most VC systems specifically address the mapping module, as the other modules

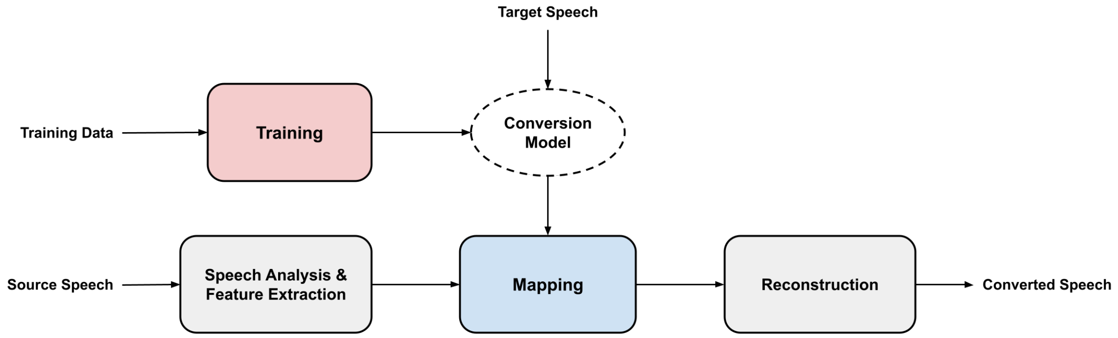


Figure 2-1: Block diagram of typical voice conversion system.

are either well-established or are significant research domains in their own right. However, as we will discuss more later, this work proposes an end-to-end VC approach, which combines the mapping and reconstruction modules together into one model. This eliminates the need for a separate vocoder and significantly streamlines the overall pipeline.

2.2.3 Traditional methods

Early research on voice conversion focused on spectrum mapping using parallel training data, where speech of the same linguistic content is available for both the source and target speaker. Popular statistical approaches used parametric methods such as Gaussian mixture models (GMMs) [104] or partial least squares regression [28], or non-parametric methods such as exemplar-based sparse representation [102, 118]. In this setting, dynamic time warping (DTW) could also be used to align the two utterances [27]. Other work also explored voice conversion using non-parallel training data [100, 15]; this is a significantly more challenging setting because of the need to establish a mapping between non-parallel source and target utterances.

In recent years, advances in deep learning have had a significant impact on voice conversion research. Not only has deep learning greatly advanced the state-of-the-art, but it has also transformed the way in which the task itself is framed. Perhaps the largest boon of deep learning has been the data-driven learning paradigm; by training on much larger amounts of data than before, VC systems have been able to achieve significant improvements in terms of voice quality and similarity to the target speaker, especially in the non-parallel data setting. Some early deep learning-based VC systems used architectures such as deep bidirectional LSTMs [99] and feedforward neural networks with KL-divergence [119]. More recent approaches have framed the conversion task as a style transfer problem, using variants of generative adversarial networks (GANs) [32, 40, 41] or vector-quantized variational autoencoders (VQ-VAEs) [112, 46]. Some methods have also leveraged the latent representations of automatic speech recognition (ASR) models to extract linguistic features from the source speech and decompose it into content and speaker information [52, 127].

Many of the voice conversion methods described above are applicable only in the *one-to-one* conversion setting; that is, they can only transform one given input speaker's voice into one specific target speaker's voice. While some of these models can perform conversion with high-fidelity, they require many hours of training data from each speaker in order to do this [94]. Some more recent methods go beyond this and are able to perform *many-to-many* voice conversion [39, 42], in which a model is able to convert voices to and from multiple speakers that have previously been seen during training.

2.2.4 Zero-shot VC

However, the setting of interest in this thesis—multi-party conversations with very little data per speaker—necessitates *zero-shot* voice conversion, an even more difficult problem. Here, the VC model must be able to convert voices to and from many different speakers who may have been previously unseen during training. Usually, the target speaker's voice is determined based on some descriptive representation of that speaker that is extracted from a single utterance. This is usually done by passing the target utterance through a "speaker encoder" network, which extracts a vector embedding that contains information about the speaker of that utterance. The speaker encoder can either be pre-trained on a speaker identification or verification task or trained jointly along with the rest of the voice conversion network.

Zero-shot voice conversion is challenging, and models that can perform it have started to appear in the literature only in recent years. The first model to demonstrate reasonable performance on the task was AutoVC [86], an autoencoder-based model that combined a pre-trained speaker encoder with carefully designed dimensionality bottleneck layers to disentangle content information from speaker information. Although it was one of the first zero-shot VC methods to be proposed, it is still quite competitive compared to many newer methods and is perhaps the model that is most often used for baseline comparisons. AutoVC has also served as the base model for a range of modifications and improvements, such as the addition of F0 information [83], mutual information-based disentangled representation learning [124], and adversarial voice style mixup for GAN-based training [50]. Other approaches for zero-shot voice conversion have used a variety of methods for disentangling the content and speaker information in speech, including adaptive instance normalization [10], activation function guidance [8], and information perturbation-based training [9].

2.2.5 End-to-end VC

Recently, some VC methods have sought to do away with the analysis-mapping-reconstruction pipeline and develop models that can be trained in an entirely end-to-end manner. The core philosophy of end-to-end models is that the modules of a learning system should be differentiable with respect to all adjustable parameters, allowing the entire system to be trained as a whole by gradient descent and back-propagation with respect to some loss [20]. Although end-to-end models have some limitations, notably with regards to interpretability and modularity, they have become popular in the context of deep learning due to their simplicity, elegance, and high performance.

In the context of voice conversion (and speech synthesis in general), end-to-end models are particularly appealing because they do not require a separate vocoder to synthesize time domain waveforms. VC models that produce spectrograms and rely on vocoders to synthesize time domain audio can have highly variable performance depending on the quality of the vocoder. Traditional vocoders such as WORLD [63] are prone to introducing artifacts such as metallic sounds into their audio. While more modern neural vocoders have fewer such issues, they can still generate poor quality audio if the spectrogram itself has flaws. VC models that generate spectrograms as an intermediate step are forced to define their training loss functions in the spectrum domain, which may not always align with human perception of audio once converted to the time domain. Consequently, many VC models are prone to producing audio that has artifacts or that sounds muffled due to oversmoothing in the spectrum.

Despite this, little prior work has been done on end-to-end voice conversion. The first end-to-end voice conversion model to be proposed was Blow [91], a normalizing flow network for non-parallel, many-to-many, raw-audio voice conversion. However, it is not able to perform zero-shot conversion, and like many other flow-based networks, has a very large number of model parameters. To the best of our knowledge, the only model currently in the literature that can perform end-to-end zero-shot VC is NVC-Net [68]. NVC-Net consists of a speaker encoder, a content encoder, a generator, and three discriminators for GAN-based training, all of which are trained jointly from scratch.

In Chapter 4, we introduce a novel approach for end-to-end voice conversion that preserves the intelligibility of the converted speech significantly better than current state-of-the-art VC models, while also achieving comparable or better voice style transfer accuracy.

Part II

Modeling

Chapter 3

Preliminaries

In this chapter, we present preliminaries, concepts, and related work that provide context for and motivate some of the key design choices in our voice conversion model.

3.1 The Source-Filter Model of Speech Production

In humans, the physical production of speech sounds involves the generation of an acoustic waveform within the vocal tract, the propagation of that waveform through the vocal tract, and its release through the speaker's mouth and nostrils [64]. Figure 3-1 shows a diagram of how speech is physically produced in this manner. There are two types of production methods:

- **Voiced speech**, which is generated by the modulation of the airstream leaving the lungs by periodic opening and closing of vocal folds in the glottis or larynx. Specifically, the vocal tract is excited by a series of nearly periodic pulses generated by the vocal cords. This production method is used for vowels and nasal consonants.

- **Unvoiced speech**, which is generated by forcing air through a narrow constriction of the vocal tract, which creates noisy turbulent airflow at the anterior end of the constriction. This produces sounds such as fricatives or unvoiced plosives.

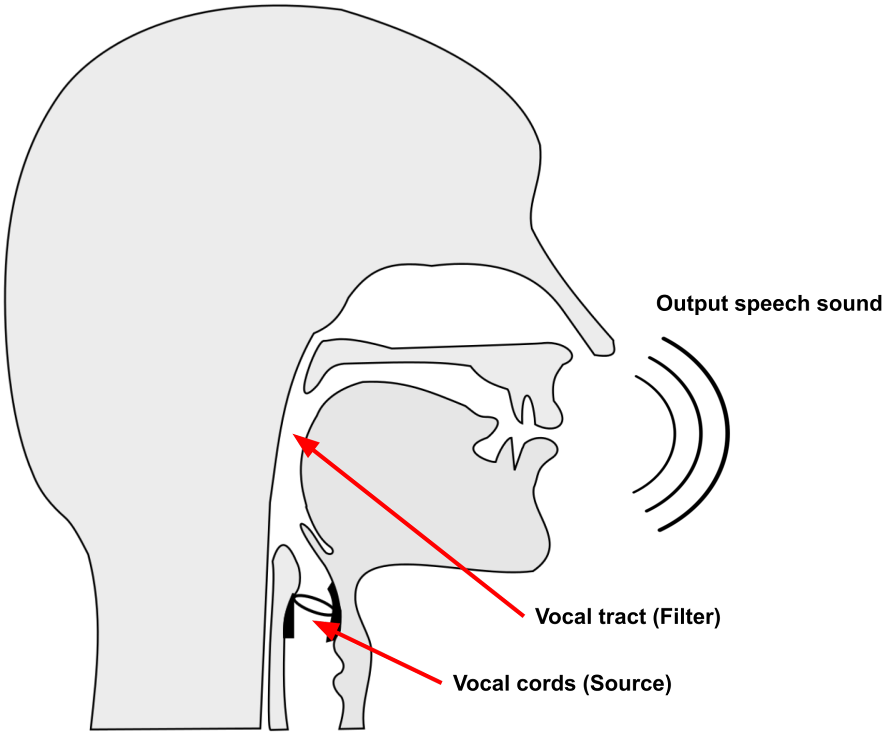


Figure 3-1: Diagram of physical speech production by the vocal cords and vocal tract.

From a signal processing point of view, the speech production process can be modeled by a linear system in which the excitation of the vocal cords (the **source**) is convolved with a representation of the vocal tract (the **filter**) [3]. Here, the excitation is modeled using either an impulse train (for voiced speech) or white noise (for unvoiced speech), represented by a signal $e(n)$ with Fourier transform $E(z)$. Meanwhile, the vocal tract can be modeled using a discrete time-varying linear filter with impulse response $h(n)$ and transfer function $H(z)$. Therefore, an output speech signal $x(n)$ and its Fourier transform $X(z)$ can be described as follows:

$$x(n) = e(n) * h(n), \tag{3.1}$$

$$X(z) = E(z)H(z), \tag{3.2}$$

where $*$ denotes the convolution operation. Figure 3-2 illustrates a block diagram of this **source-filter model** of speech production.

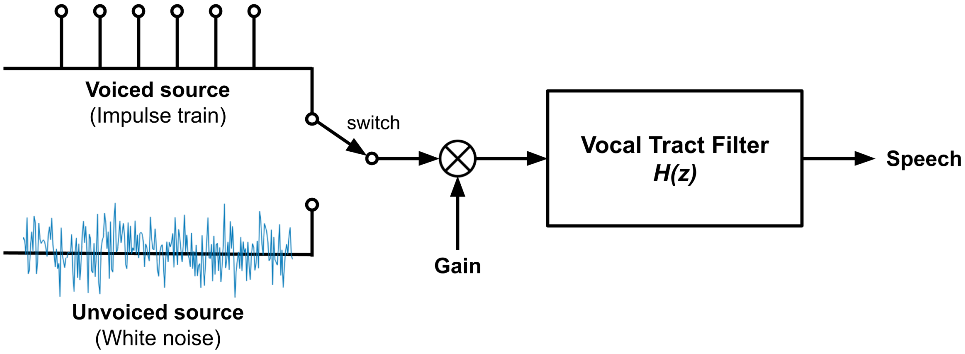


Figure 3-2: Block diagram of the source-filter model of voice production.

The transfer function of a linear system can always be represented by its poles and zeros; however, for non-nasal voiced speech sounds, the transfer function of the vocal tract can be modeled with no zeros provided that the order of pole coefficients is sufficiently large [17]. Therefore, the vocal tract can be represented using an all-pole filter whose coefficients are determined over time through linear predictive coding.

Physically, the vocal tract can be viewed as an acoustic tube of varying diameter at different points. Depending on the shape of the acoustic tube, a sound wave traveling through it will be reflected in such a way that interferences will create different weighted magnitudes of frequencies across the frequency spectrum [4]. At a given point in time, these make up the **spectral envelope** of the voice, which determines the specific phoneme that is produced via resonances called formants. The source-filter model provides a way of modeling the spectral envelope: it can be approximated by the transfer function of the filter, $H(z)$ [3]. Figure 3-3 shows an example of an utterance's spectral envelope extracted from an analysis window.

In the context of the information in speech described in Section 2.2.1, we can consider the excitation of the vocal cords $E(z)$ to contain some of the *speaker* information in a spoken utterance. Indeed, $E(z)$ contains information on a voice's fundamental frequency (F0) as well as its harmonic frequencies. Meanwhile, we can consider the spectral envelopes and formants $H(z)$ over time to contain a significant portion of the *content* information of an utterance. In voice conversion, one of the required steps is to disentangle the content information of an utterance from its speaker information.

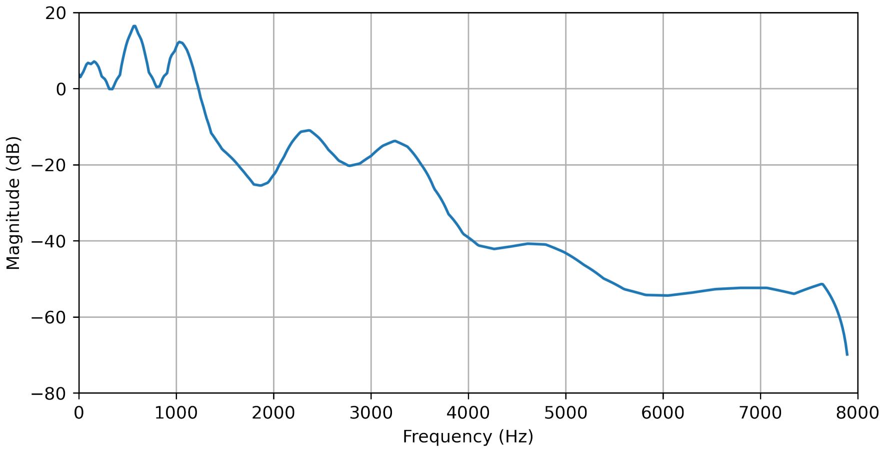


Figure 3-3: The spectral envelope of an utterance at a given analysis frame. Peaks in the envelope represent formant frequencies.

As we see here, one way of doing this is to separate $H(z)$ from $E(z)$ by performing deconvolution.

It should be noted that $H(z)$ still contains a significant amount of speaker information on its own. Intuitively, this is because different speakers have different vocal tract shapes, which causes variations in the way in which spectral envelopes and formants are shaped even when the same phonemes are being pronounced. Therefore, deconvolution of $H(z)$ and $E(z)$ is not expected to fully disentangle speaker and content information on its own.

3.1.1 Deconvolution in the cepstrum

Recall from Equation 3.1 that a speech signal $x(n)$ can be expressed as a convolution between an excitation signal $e(n)$ and the impulse response of the vocal tract filter $h(n)$. In the frequency domain, this convolution becomes equivalent to the multiplication of their respective Fourier transforms, as shown in Equation 3.2. Taking the logarithms of the absolute values of the Fourier transforms to compute the log magnitude spectra converts the multiplication operation to addition:

$$\begin{aligned} \log|X(z)| &= \log|E(z)H(z)| && (3.3) \\ &= \log|E(z)| + \log|H(z)|. && (3.4) \end{aligned}$$

If we apply a Fourier transform (in practice, actually a discrete cosine transform (DCT) since the log magnitude spectrum only has real components) to the above, we obtain a frequency distribution of the fluctuations in the curve of the spectrum, called the **cepstrum**[1] (C):

$$C = \mathrm{DCT}(\log |X(z)|) \tag{3.5}$$

$$= \mathrm{DCT}(\log |E(z)|) + \mathrm{DCT}(\log |H(z)|). \tag{3.6}$$

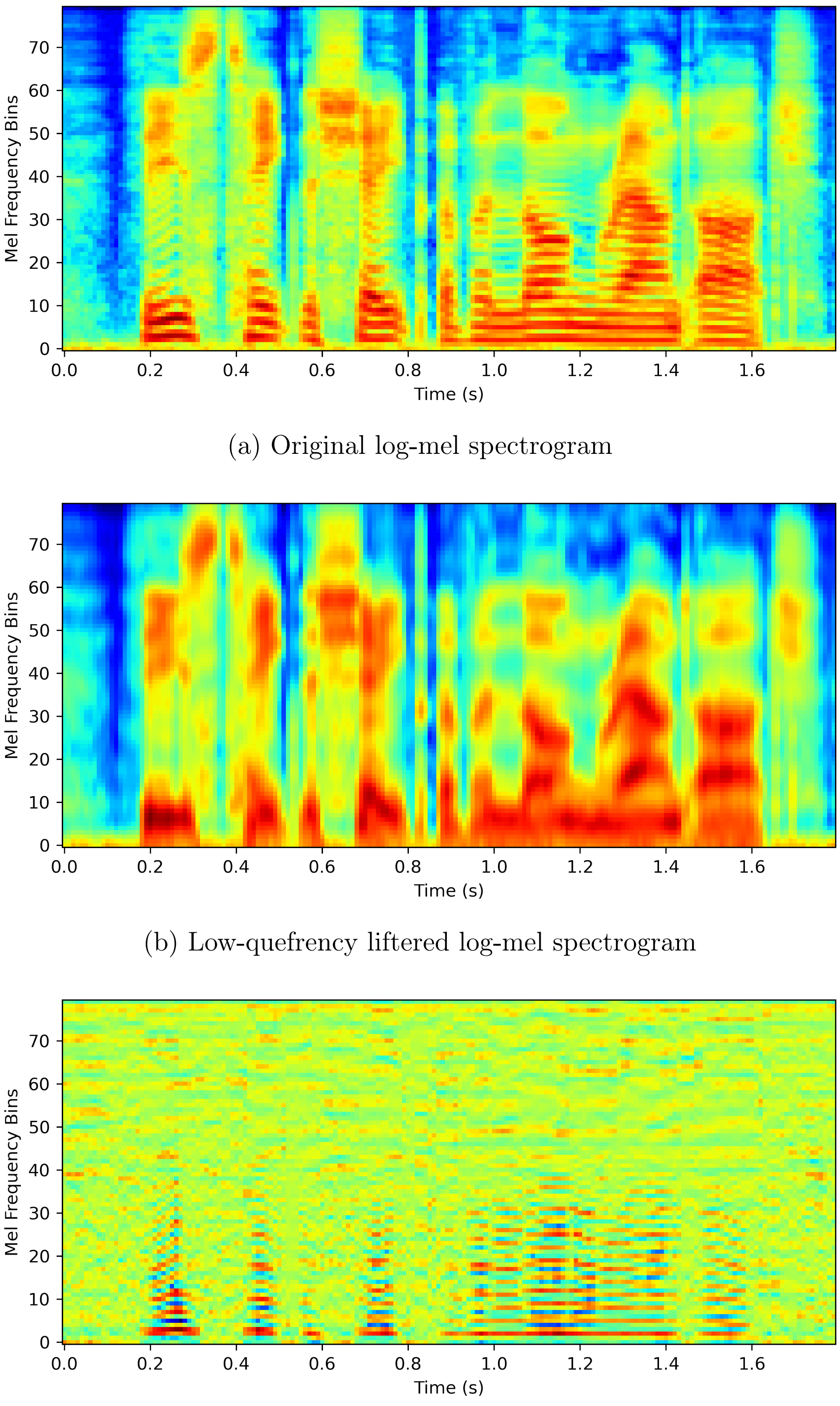
If we assume that the source (excitation) spectrum has only rapid fluctuations (since the excitation signal is a stable, regular oscillation), its contribution to the cepstrum will be concentrated in the higher *quefrency*[2] bins of C. Conversely, the filter (vocal tract) will contribute slow fluctuations to the spectrum of X and will be concentrated in the lower quefrency bins.

Therefore, the separation of $E(z)$ and $H(z)$ becomes straightforward: we simply have to perform *liftering*[3] and select the desired quefrency region by multiplying the entire cepstrum by a window at the appropriate position. Low-quefrency liftering, where the quefrency coefficients below a certain point are extracted, allows us to obtain the vocal tract characteristics in the quefrency domain. High-quefrency liftering, the opposite, allows us to obtain the excitation characteristics. Once we have performed liftering, it is a simple matter of performing the inverse DCT to obtain the deconvolved spectral envelope and excitation. Figure 3-4 illustrates the results of performing low-quefrency and high-quefrency liftering on a sample log-mel spectrogram.

[1] From flipping the first part of the word *spectrum*.

[2] The cepstrum equivalent of *frequency*.

[3] *Filtering* in the cepstrum domain.



(a) Original log-mel spectrogram

(b) Low-quefrency liftered log-mel spectrogram

(c) High-quefrency liftered log-mel spectrogram

Figure 3-4: A sample log-mel spectrogram (a) and the results of performing (b) low-quefrency and (c) high-quefrency liftering on it. Note that (b) captures the spectral envelope and formants of the utterance, while (c) captures the F0 and its harmonic frequencies.

3.2 Deep Generative Models for Speech Synthesis

Modeling audio is a challenging problem because of the high temporal resolution of the data (sampling rates are usually at least 16 kHz and can go up to 48 kHz or higher) and the presence of structure at different time scales with both short- and long-term dependencies. This is especially difficult in the context of speech synthesis, as models must be able to accurately capture this structure and generate the samples of a time domain waveform while maintaining high perceptual fidelity.

Generally, speech synthesis models operate by taking a lower-resolution intermediate audio representation as their input and reconstructing the corresponding time domain audio. As such, these models can be considered **vocoders**. The intermediate representations are usually chosen to be easier to model than raw audio while preserving enough information to allow accurate inversion back to the time domain. Mel spectrograms [19, 92] are perhaps the most commonly used type of intermediate representation, although other representations such as aligned linguistic features [73] can also be used.

In recent years, deep generative neural network models (so-called *neural vocoders*) have achieved great success in speech synthesis, demonstrating significantly improved performance compared to traditional signal processing methods [23, 63]. They can largely be classified into three families: autoregressive models, non-autoregressive models, and generative adversarial network (GAN)-based models.

3.2.1 Autoregressive models

WaveNet [73] was one of the first deep generative models to demonstrate success in speech synthesis. It is a fully convolutional model that uses dilated causal convolutions to produce speech samples in an autoregressive manner, conditioned on linguistic features that are temporally aligned with the raw audio. WaveRNN [37] was subsequently introduced as a faster model based on a simple, single-layer recurrent neural network; it introduced various techniques such as weight sparsification and subscale generation to improve synthesis speed.

However, inference with autoregressive models is fundamentally quite slow because audio samples must be generated sequentially, which makes them impractical for real-time applications.

3.2.2 Non-autoregressive models

To address these issues, many non-autoregressive models were subsequently proposed to generate waveforms more quickly. These models can be orders of magnitude faster than their autoregressive counterparts because they are highly parallelizable and can fully exploit modern hardware such as GPUs and TPUs. Some types of non-autoregressive models, such as Parallel WaveNet [72] and ClariNet [78], utilize knowledge distillation, in which a trained auto-regressive decoder is distilled into a flow-based convolutional student model. The student model is then able to perform inference much more quickly than the teacher model while achieving a similar level of performance. Other non-autoregressive models, such as WaveGlow [81] and WaveFlow [79], utilize flow-based methods, using autoregressive and inverse autoregressive flows to represent high capacity generative flows for audio. While flow-based methods enable fast speech generation at inference time, they tend to have very large model sizes that make them impractical for applications with constrained memory budgets.

3.2.3 GAN-based models

More recently, generative adversarial networks (GANs) [22] have become a popular way of training speech synthesis models. By designing generator and discriminator losses strategically, it is possible to train a model to synthesize high quality audio with a compact and fast generator architecture.

The performance of GAN-based models is largely dependent on the ability of their discriminators to discern real and fake generated samples. Therefore, many of the advances in GAN-based speech synthesis have come as a result of more clever and sophisticated discriminator designs. MelGAN [49] utilized a collection of discriminators that evaluated the generated audio at multiple timescales to determine their authenticity. Parallel WaveGAN [122] introduced a multi-resolution short-time Fourier transform (STFT) loss to stabilize GAN training. More recently, HiFi-GAN [47] introduced a discriminator design that evaluates synthesized audio at various periods in the time domain, and UnivNet [35] introduced a design that performs an analogous role for various spectral resolutions in the frequency domain.

GAN-based vocoders have achieved state-of-the-art results in speech synthesis, being able to generate audio with clarity and naturalness that approach ground truth speech. Because of their compact generator architectures, they are also able to perform inference extremely quickly—often hundreds of times faster than real-time on GPUs and faster than real-time even on CPUs.

3.3 Location-Variable Convolutions

Most of the aforementioned speech synthesis models are implemented using a WaveNet-like generator network, in which mel spectrograms are used as conditioning features and dilated causal convolutions are applied to capture the long-term dependencies of a waveform. This necessitates a large number of convolution kernels in order to properly capture the many time-dependent features that arise in speech, since the same convolutional kernel weights must be used for all audio frames. However, in a traditional linear prediction vocoder [3], the coefficients for the all-pole linear filter vary depending on the conditioning acoustic features of the analysis frame. What if a network could have variable kernel coefficients depending on the conditioning features? Such a network could then be able to model long-term dependencies in waveforms much more efficiently than fixed-kernel methods. Inspired by these ideas, [126] recently introduced **location-variable convolutions (LVCs)**, in which different convolutional kernel weights are used to model different intervals in the input sequence depending on the corresponding "local" sections of a conditioning sequence such as a mel spectrogram.

Formally, let the input sequence to the convolution operation be $\mathbf{x} = \{x_1, x_2, ..., x_n\}$, and define the local conditioning sequence as $\mathbf{h} = \{h_1, h_2, ..., h_m\}$. Each element in

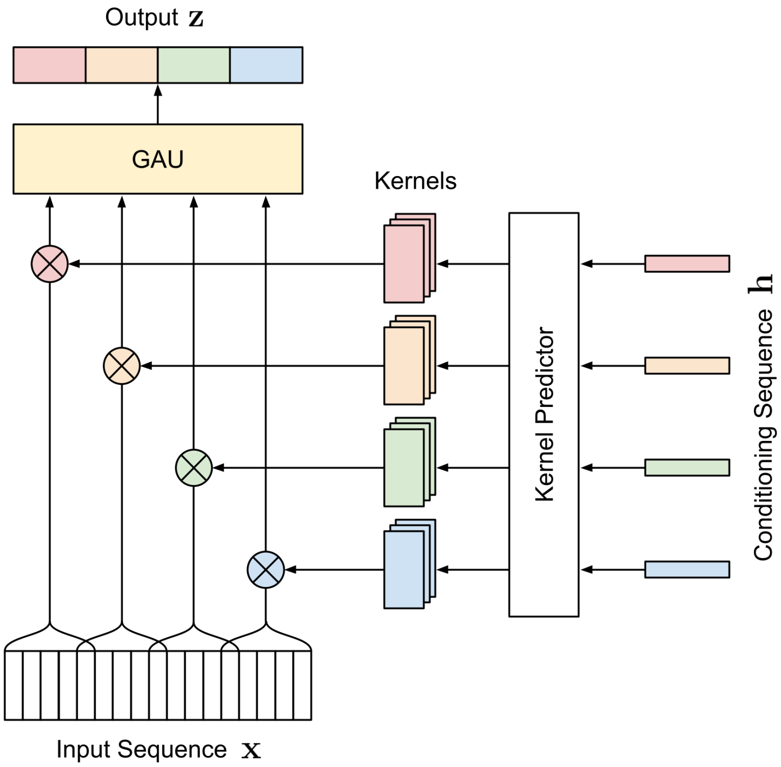

Output $\mathbf{z}$
GAU
Kernels
Kernel Predictor
Conditioning Sequence $\mathbf{h}$
Input Sequence $\mathbf{x}$

Figure 3-5: Diagram of the location variable convolution (LVC) process.

the local conditioning sequence is associated with a given continuous interval in the input sequence. LVCs utilize a **kernel predictor network** (KPNet) whose purpose is to predict the weights of convolution kernels given a local conditioning sequence. Therefore, each element in the local conditioning sequence determines the specific convolution kernels that are applied to its associated input sequence interval. In other words, each interval of the input sequence has a different convolution operation performed on it depending on the temporally associated section of the conditioning sequence. The final output sequence $\mathbf{z}$ is produced by splicing together the convolution results from each processing interval after passing them through a gated activation unit (GAU) [111]. Figure 3-5 illustrates a diagram of the role of the kernel predictor network in an LVC layer.

The operations done by an LVC layer can be expressed by the following equations:

$$\{\mathbf{x}_{(i)}\}_m = \text{split}(\mathbf{x}), \tag{3.7}$$

$$\{\mathbf{W}^f_{(i)}, \mathbf{W}^g_{(i)}\}_m = \text{KPNet}(\mathbf{h}), \tag{3.8}$$

$$\mathbf{z}_{(i)} = \tanh(\mathbf{W}^f_{(i)} * \mathbf{x}_{(i)}) \odot \sigma(\mathbf{W}^g_{(i)} * \mathbf{x}_{(i)}), \tag{3.9}$$

$$\mathbf{z} = \text{concat}(\mathbf{z}_{(i)}), \tag{3.10}$$

where $\mathbf{x}_{(i)}$ denotes the intervals of the input sequence associated with h_i, and $\mathbf{W}^f_{(i)}$ and $\mathbf{W}^g_{(i)}$ denote the filter and gate convolution kernels for $\mathbf{x}_{(i)}$, respectively. $*$ denotes the convolution operation and $\odot$ denotes element-wise multiplication.

Intuitively, LVCs have more powerful capabilities for modeling long-term dependencies in audio because they can flexibly generate kernel weights that correspond in a more customized way to different conditioning sequences.

Chapter 4

LVC-VC: End-to-End Zero-Shot Voice Conversion

We now bring together the preliminaries established in Chapter 3 and introduce **Location-Variable Convolution-based Voice Conversion (LVC-VC)**, our proposed model for end-to-end zero-shot voice conversion.

LVC-VC is based on UnivNet [35], a neural vocoder that combines location-variable convolutions in its generator with a variety of discriminators for GAN-based training. We chose to use UnivNet's base architecture and training strategy because of its capabilities for generating very high-fidelity audio. It was shown to outperform other state-of-the-art GAN-based vocoders in objective and subjective evaluations, regardless of whether the speakers of input spectrograms had been seen or unseen during training. In addition, it is able to achieve very fast inference speeds even compared to other GAN-based methods, which allows for real-time audio synthesis. As with many other vocoders, UnivNet takes a log-mel spectrogram as its input feature and reconstructs time domain audio corresponding to that spectrogram. To adapt the model to perform voice conversion, however, we made several modifications to the original model's input features, architecture, and training strategy.

Most zero-shot VC models trained on non-parallel data use a self-supervised training strategy with some form of self-reconstruction loss. At a high-level, the idea is to decompose a given training utterance into separate speaker and content embeddings, and then recombine them using a decoder to reconstruct the original signal. At inference time, conversion can simply be performed by extracting the content embedding from the source utterance and combining it with the speaker embedding from the target utterance. The hope is that the model will learn to combine content and speaker embeddings in a coherent way regardless of where each of them comes from, thereby synthesizing audio that corresponds to the content of the source utterance spoken in the target speaker's voice. Given this, most models are composed of a content encoder, speaker encoder, and decoder. Some previous VC methods have trained all of these components jointly from scratch [10, 8, 68], while others have used a speaker encoder that was pre-trained on a separate speaker verification task [86, 83].

LVC-VC consists of a generator G, a speaker encoder E_s, and a set of discriminators for GAN-based training. Like most other zero-shot VC models, it is trained using a self-reconstruction paradigm. However, because the model is based on a neural vocoder, it does not include an explicit content encoder or decoder in its architecture. Rather, we feed a specific set of carefully designed input features that already have some amount of information disentanglement into the model's kernel predictor network. Then, the kernel predictor is tasked with combining the information from the various features and passing it to the generator to perform audio synthesis. While this strategy removes some degree of interpretability, it significantly streamlines the model's overall structure and largely removes the difficult task of teaching the model to perform proper disentangled representation learning.

4.1 Input Features

4.1.1 Content features

In voice conversion, the source utterance is responsible for providing content information. Let the source utterance in the time domain be $\mathbf{x}$ and its log-mel spectrogram be $\mathbf{X}$. We perform low-quefrency liftering on $\mathbf{X}$ to extract the spectral envelope of the utterance $\mathbf{H}$. $\mathbf{H}$ serves as the primary content feature that is fed into the model, and is intended to contain the content information of $\mathbf{X}$ but with the excitation of the vocal cords (the harmonics) removed.

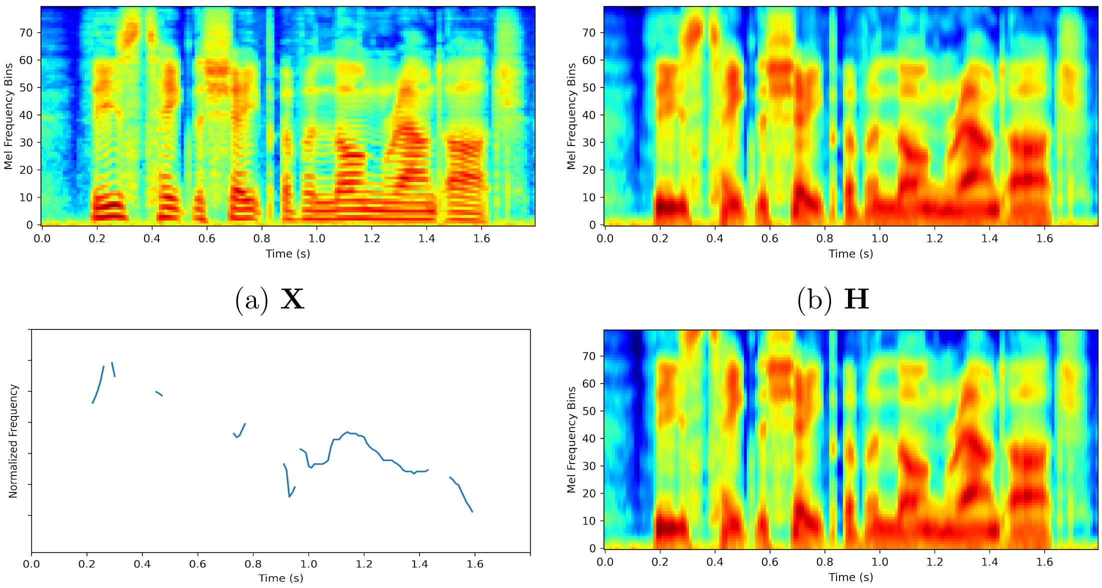
However, recall from Section 3.1 that a low-quefrency liftered spectrogram still contains a significant amount of speaker information. To prevent the model from using this residual speaker information in the feature, we randomly warp $\mathbf{H}$ by stretching or compressing it along the frequency axis during training. The warping is done via linear interpolation between frequency bin values, and we denote the warped version of the feature $\mathbf{H}'$. We found that this step removes most of the residual speaker information in low-quefrency liftered log-mel spectrograms while still preserving the content information represented in the original spectral envelope. A similar information perturbation strategy was used by a speech decomposition and synthesis model in [9] to constrain the information that would be extracted from certain input features.

We also compute the normalized F0 contour of $\mathbf{x}$ and use it as an additional input feature for the model. Specifically, we use the per-frame normalized quantized log F0 feature $\mathbf{p}_{\text{norm}}$ that was introduced and used previously in [83] and [84]. While an utterance's F0 contour does not encode any linguistic content information, it does contain meaningful information about the pitch, prosody, and intonation of an utterance over time (i.e., how it was said). Therefore, we broadly categorize this feature as content information since it encodes aspects of the source utterance that we wish to preserve in the output.

To compute $\mathbf{p}_{\text{norm}}$, we first extract the log F0 from all of a speaker's voiced samples using a pitch tracking algorithm [56]. Then, we compute the speaker's log F0 mean μ and variance σ^2. We use the same analysis window size and and hop as when computing $\mathbf{X}$ to make sure that the number of extracted F0 frames matches up with the number of spectrogram frames. Then, for each voiced frame, we normalize the raw log F0 p_{raw} as follows:

$$p_{\text{norm},n} = \frac{p_{\text{raw},n} - \mu}{4\sigma} + \frac{1}{2}, \tag{4.1}$$

where n denotes the index of an analysis frame. This operation roughly constrains



(a) $\mathbf{X}$

(b) $\mathbf{H}$

(c) $\mathbf{p}_{\text{norm}}$

(d) $\mathbf{H}'$

Figure 4-1: Various content-related input features for LVC-VC extracted from a sample utterance $\mathbf{x}$. (a) Log-mel spectrogram $\mathbf{X}$, (b) Low-quefrency liftered log-mel spectrogram $\mathbf{H}$, (c) Normalized log F0 contour extracted from $\mathbf{x}$, (d) Warped low-quefrency liftered log-mel spectrogram $\mathbf{H}'$ (stretched by a factor of 1.15).

values of p_{norm} to be within the range $[0, 1]$; any values falling outside this range are clipped. We then quantize the range $[0, 1]$ into 256 bins and one-hot encode the p_{norm} values. Finally, we add another bin to represent unvoiced frames, resulting in a 257-dimensional one-hot encoded feature for each frame. The concatenation of these features across all frames then becomes $\mathbf{p}_{\text{norm}}$, with dimensions $(257, N)$:

$$\mathbf{p}_{\text{norm}} = [p_{\text{norm},1}; p_{\text{norm},2}; ...; p_{\text{norm},N}]. \tag{4.2}$$

Figure 4-1 shows visualizations of the various content features. The idea is that during training, LVC-VC should learn to extract the appropriate information from the speaker features (see Section 4.1.2) and combine them with $\mathbf{H}'$ and $\mathbf{p}_{\text{norm}}$ so as to "un-warp" $\mathbf{H}'$, "un-normalize" $\mathbf{p}_{\text{norm}}$, and add the appropriate excitation harmonics to reconstruct the original signal. At inference time, the model should use information about the target speaker to perform the "un-warping" and "un-normalizing" in a way that causes the generated audio to sound like the target speaker's voice.

4.1.2 Speaker features

For speaker-related conditioning features, we use embeddings extracted from a speaker encoder E_s that has been pre-trained on a speaker recognition task. The architecture and training details of E_s are described more in-depth in Section 4.2.2. We denote speaker embeddings using the variable s. For an utterance $\mathbf{x}$ with log-mel spectrogram $\mathbf{X}$, the speaker embedding is then:

$$s = E_s(\mathbf{X}). \tag{4.3}$$

A good speaker encoder should produce embeddings that are close together in the latent space for utterances from the same speaker, regardless of the utterances' content. Conversely, it should produce embeddings that are farther apart in the latent space for utterances spoken by different speakers. Furthermore, a speaker embedding should encode information about its speaker's vocal characteristics so that embeddings from unseen speakers can be used for zero-shot voice conversion.

In addition to the speaker embedding, we also include the quantized median log F0 value of a speaker as an additional conditioning feature. This is computed as follows. We extract the log F0 from all of a speaker's voiced speech samples using the same pitch tracking algorithm as for $\mathbf{p}_{\text{norm}}$ above and compute the median for a speaker. Then, we quantize the range log(65.4) Hz to log(523.3) Hz (corresponding to the notes 'C2' and 'C5') into 64 bins and and one-hot encode the median log F0 values. As before, any F0 values falling outside the quantized range are clipped. This results in a 64-dimensional vector m that encodes a speaker's F0 information.

4.2 Model Architecture

4.2.1 Generator

The generator G is largely based on the generator in UnivNet, specifically the UnivNet-c16 variant, which has a channel size of 16 in each of its convolutional layers. Figure

4-2 shows a diagram of the overall architecture. In total, the generator contains around 4.5 million parameters. It is a fully convolutional neural network that takes random noise $\mathbf{z}$ as an input sequence and the content and speaker features described in Section 4.1 as conditions, and outputs a raw audio waveform $\hat{\mathbf{x}}$. Its main body consists of a series of 1D transposed convolutional layers to upsample the input noise sequence $\mathbf{z}$, which is specified to have the same length as the low-quefrency liftered log-mel spectrogram $\mathbf{H}$. In our experiments, spectrograms are at a 256× lower resolution compared to raw audio. Therefore, there are three transposed convolutional layers with upsampling factors of 8×, 8×, and 4× to perform the total 256× upsampling. This results in the output waveform $\hat{\mathbf{x}}$ having the same length as the source waveform $\mathbf{x}$ from which $\mathbf{H}$ was extracted.

Each transposed convolutional layer is followed by a stack of four residual blocks that gradually transform the noise sequence into the final waveform as it is passed through them. Each residual block consists of a dilated 1D convolution, a 1D location-variable convolution (LVC), and a gated activation unit [111]. The four dilated convolutions in each stack have dilation factors of [1, 3, 9, 27]. Leaky ReLU [54] with $\alpha = 0.2$ is used as the activation before the dilated convolutions and LVCs. The kernels of the LVC layers are determined by kernel predictor networks that take the conditioning features $\mathbf{H}$, $\mathbf{p}_{\text{norm}}$, s, and m as input (Figure 4-3). Each residual stack has its own kernel predictor network, for a total of three kernel predictors. Each kernel predictor consists of a residual stack of 1D convolutions with Leaky ReLU activations ($\alpha = 0.2$), and simultaneously predicts the kernels of all of the LVC layers in the stack that it is associated with.

The output waveform is thus a result of feeding the input noise sequence and all of the conditioning features through the generator:

$$\hat{\mathbf{x}} = G(\mathbf{z}, \mathbf{H}, \mathbf{p}_{\text{norm}}, s, m). \tag{4.4}$$

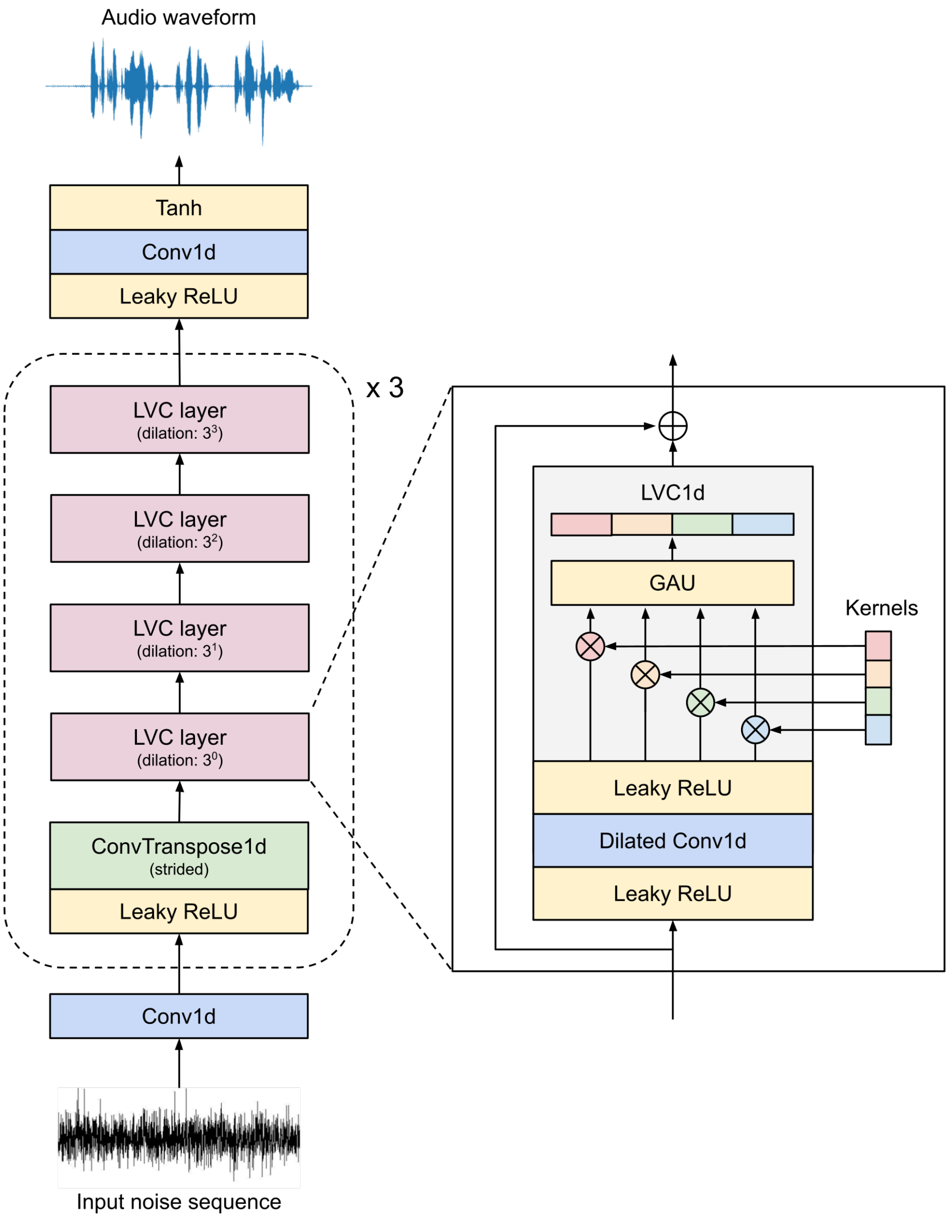


Figure 4-2: LVC-VC generator architecture. Kernels for LVC layers come from the kernel predictor network.

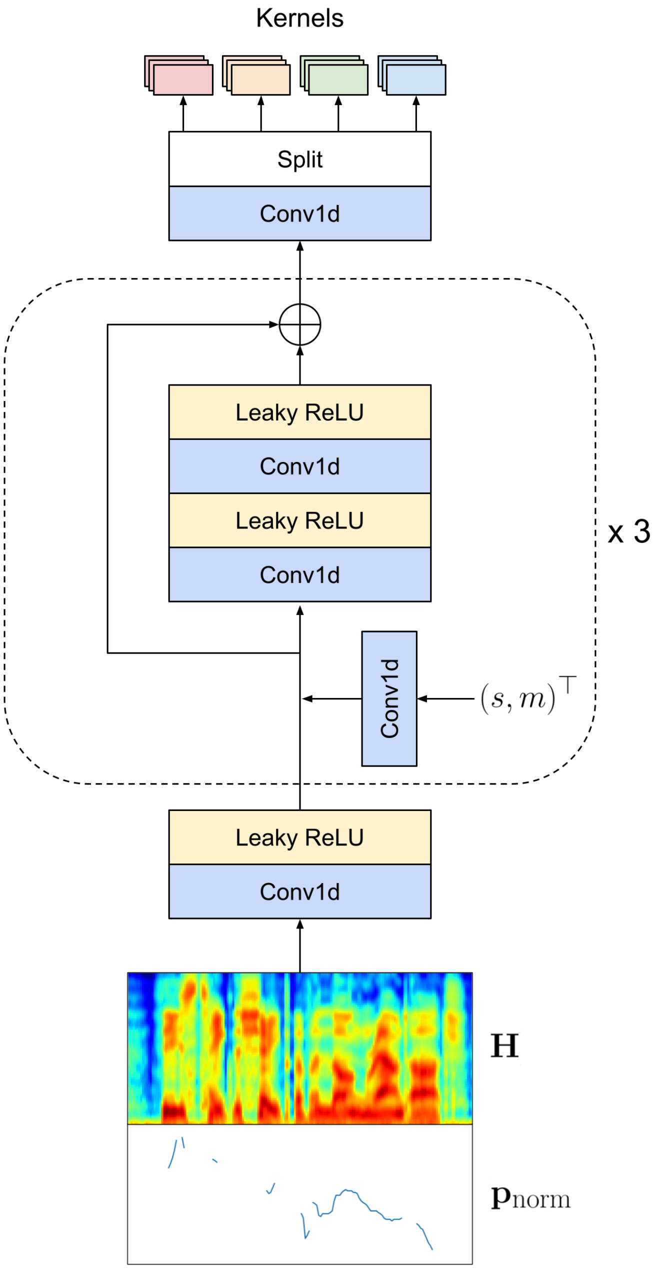


Figure 4-3: Kernel predictor network for LVC-VC.

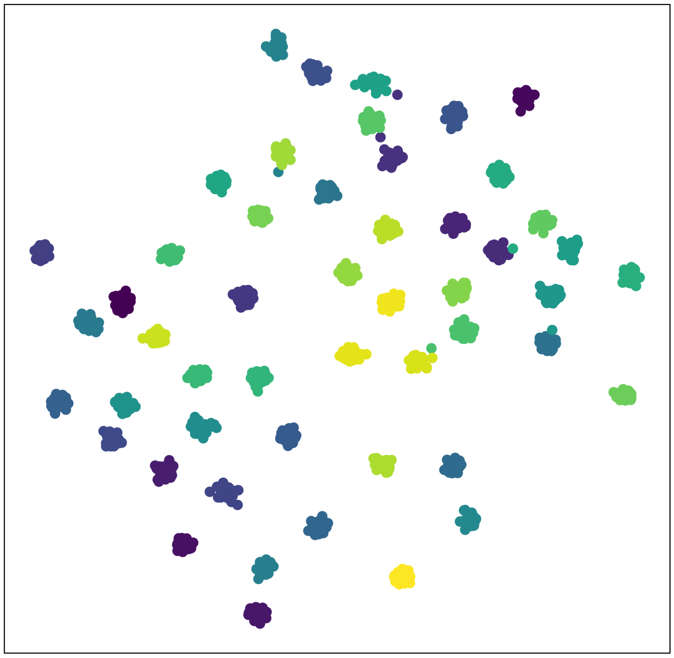


Figure 4-4: t-SNE visualization of embeddings extracted from the utterances of 50 speakers in the VCTK dataset using E_s. Each color denotes a different speaker.

4.2.2 Speaker encoder

For the speaker encoder E_s, we use the Fast ResNet-34 speaker recognition model from [11]. The model was pre-trained using angular prototypical loss on the development set of the VoxCeleb2 dataset [12] and uses self-attentive pooling [7] to aggregate frame-level features into an utterance-level representation. It is based on the original ResNet-34 architecture [26], but has only one-quarter of the channels in each residual block and earlier strides in order to reduce the computational complexity. The model takes 40 dimensional log-mel spectrograms as input and outputs speaker embeddings of dimension 512.

Although very fast and lightweight (about 1.4 million parameters), the model achieves an impressive equal error rate (EER) of 2.18% on a speaker verification task for the VoxCeleb1 test set [65]. We chose to use it because of its combination of efficiency and high quality speaker representations. Figure 4-4 shows a t-SNE visualization of speaker embeddings extracted from the utterances of 50 speakers in the VCTK dataset [120] using E_s.

4.2.3 Discriminators

In addition to the use of LVCs in its architecture, one of the key components of UnivNet that helps it generate such high quality audio is the design of its discriminators for GAN-based training. UnivNet utilizes two discriminators, a multi-resolution spectrogram discriminator (MRSD) and a multi-period waveform discriminator (MPWD), which we also use for training LVC-VC.

Multi-resolution spectrogram discriminator (MRSD)

During training, the purpose of the MRSD is to evaluate a synthesized audio waveform at multiple frequency resolution scales and make a decision as to whether the waveform is real audio or not. Hence, the MRSD actually consists of M sub-discriminators, each of which evaluates and makes a decision on the accuracy of the generated audio at a given spectral resolution. The sub-discriminators of the MRSD compute M linear magnitude spectrograms from the true audio $\mathbf{x}$ and synthesized audio $\hat{\mathbf{x}}$ during self-reconstructive training using M short-time Fourier transform (STFT) parameter sets, $\{\mathrm{FT}_m(\cdot)\}_{m=1}^{M}$. Here, $\mathrm{FT}_m(\cdot)$ denotes the Fourier transform performed by the m-th sub-discriminator. Each STFT parameter set consists of: (number of points for the Fourier transform, window length (in seconds), hop length (in seconds)). Formally, the sub-discriminators compute the following:

$$\{\mathbf{s}_m = |\mathrm{FT}_m(\mathbf{x})|, \hat{\mathbf{s}}_m = |\mathrm{FT}_m(\hat{\mathbf{x}})|\}_{m=1}^{M}. \tag{4.5}$$

By employing multiple spectrograms with various temporal and spectral resolutions to analyze audio, the MRSD's objective is to determine whether a given waveform's spectral characteristics appear to be real or not. In doing so, it is able to induce the generator to produce higher fidelity audio. In our experiments, $M = 3$ and the STFT parameter sets for the sub-discriminators were [(512, 0.025, 0.005), (1024, 0.05, 0.01), (256, 0.01, 0.002)].[1]

[1] The number of Fourier transform points here for each window length are appropriate for audio at 16 kHz sampling rates, which we used throughout this work. At other sampling rates, a different number of Fourier transform points may need to be used with analysis windows of the same length.

The MRSD's sub-discriminators all follow the same basic architecture, which is inspired by the multi-scale waveform discriminator used in [49]. It consists of strided 2D convolutions followed by Leaky ReLU activations with $\alpha = 0.2$.

Multi-period waveform discriminator (MPWD)

The MPWD, originally introduced in [47], is also a mixture of sub-discriminators, each of which takes as input equally spaced samples of a time domain audio waveform at a different period p and makes a decision as to whether the audio is real or not. Each sub-discriminator consists of a stack of strided 2D convolutional layers with Leaky ReLU activations ($\alpha = 0.2$). The periods are set to the prime numbers $p \in [2, 3, 5, 7, 11]$ in order to avoid overlaps in analysis between the sub-discriminators as much as possible.

Specifically, given a 1D raw audio signal of length T, the audio is reshaped into a 2D array of width p and height T/p. The reshaped audio is then fed through the sub-discriminator corresponding to the period p, which makes a decision as to whether the audio is real or fake. Collectively, the sub-discriminators of the MPWD are designed to model and capture implicit structures in the periodic patterns of audio at multiple temporal resolutions, thereby guiding the generator to synthesize more realistic waveforms.

4.3 Training

Recall that we use a speaker encoder E_s that has already been pre-trained to extract embeddings with some form of speaker information. Therefore, to train LVC-VC, we keep the weights of E_s fixed and only train the generator and discriminators.

4.3.1 Loss functions for self-reconstruction

As mentioned previously, LVC-VC is trained primarily using a self-reconstruction paradigm. In this setting, both the source and the target utterances are set to be the same, and the model's objective is to reconstruct the original utterance as closely as possible. Let an input utterance for training be $\mathbf{x}$ and the associated conditioning features be $\mathbf{H}', \mathbf{p}_{\text{norm}}, s$, and m (recall that we randomly warp $\mathbf{H}$ during self-reconstructive training only). Then, the reconstructed output is produced by $\hat{\mathbf{x}} = G(\mathbf{z}, \mathbf{H}', \mathbf{p}_{\text{norm}}, s, m)$. We use some of the same loss functions that are used to train the baseline UnivNet vocoder, described below.

In addition to the GAN losses defined by the discriminators in Section 4.2.3, multi-resolution STFT loss [122] is used as an auxiliary training criterion. It is made up of the sum of multiple spectrogram losses computed using various STFT parameter sets. The full auxiliary loss $\mathcal{L}_{\text{aux}}$, which is comprised of the spectral convergence loss $\mathcal{L}_{\text{sc}}$ and log STFT magnitude loss $\mathcal{L}_{\text{mag}}$, is defined as follows:

$$\mathcal{L}_{\text{sc}}(\mathbf{s}, \hat{\mathbf{s}}) = \frac{\|\mathbf{s} - \hat{\mathbf{s}}\|_F}{\|\mathbf{s}\|_F}, \tag{4.6}$$

$$\mathcal{L}_{\text{mag}}(\mathbf{s}, \hat{\mathbf{s}}) = \frac{1}{N}\| \log \mathbf{s} - \log \hat{\mathbf{s}}\|_1, \tag{4.7}$$

$$\mathcal{L}_{\text{aux}}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{M}\sum_{m=1}^{M} \mathbb{E}_{\mathbf{x},\hat{\mathbf{x}}}\Big[\mathcal{L}_{\text{sc}}(\mathbf{s}_m, \hat{\mathbf{s}}_m) + \mathcal{L}_{\text{mag}}(\mathbf{s}_m, \hat{\mathbf{s}}_m)\Big]. \tag{4.8}$$

Here, N denotes the number of frames in the spectrogram and $\|\cdot\|_F$ and $\|\cdot\|_1$ denote the Frobenius and L1 norms, respectively. M is the number of MRSD sub-discriminators. $\mathbf{s}$ and $\hat{\mathbf{s}}$ are defined as in Equation 4.5, and each m-th $\mathcal{L}_{\text{sc}}$ and $\mathcal{L}_{\text{mag}}$ reuse the $\mathbf{s}_m$ and $\hat{\mathbf{s}}_m$ that are used for the m-th MRSD sub-discriminator.

4.3.2 Loss functions for speaker similarity

Through self-reconstructive training, voice conversion models are meant to learn how to combine speaker and content information in a coherent way, thereby synthesizing audio that corresponds to the content of the source utterance spoken in the target speaker's voice. However, self-reconstructive training on its own does not explicitly force the converted audio to take on the vocal characteristics of the target speaker. In this setting, the amount of voice style transfer that can actually happen is highly dependent on the quality of the speaker encoder E_s—that is, whether its embeddings indeed encode enough of a speaker's vocal identity—and how well the VC model learns to utilize the information in that embedding. Indeed, training a model to properly disentangle and utilize the content and speaker information from an utterance is difficult. There are cases in which content embeddings end up containing some speaker information that has leaked through, or where they do not preserve all of the linguistic information from the source utterance. This can result in converted speech that still sounds like the source speaker or has unclear, garbled pronunciation of words. These issues can be especially noticeable when conversion is being performed on previously unseen speakers in the zero-shot setting.

LVC-VC avoids the second of these issues by means of its input feature design. Because it utilizes the full spectral envelope of the source utterance, content information is well-preserved in the output audio no matter what. However, we found during our experiments that the spectral envelope still carried some amount of speaker information through the synthesis, even with the warping strategy to perturb this information during training. This sometimes caused converted audio to maintain some aspects of the source speaker and not be fully transformed into the target speaker's voice.

Therefore, we utilize an additional loss which induces LVC-VC to generate audio that more closely matches the characteristics of the target speaker. We call this loss the speaker similarity criterion (SSC). To implement it, we make LVC-VC generate voice-converted audio by using speaker features that are different from those of a source utterance from which the content features are extracted. Then, the converted utterance is explicitly guided to sound more like the target speaker's voice.

Formally, let the original utterance used for self-reconstructive training be $\mathbf{x}_0$ and its associated features be $(\mathbf{H}_0, \mathbf{p}_{\text{norm},0}, s_0, m_0)$. For each reconstructive training sample, we sample N more utterances from different speakers $\mathbf{x}_1, ..., \mathbf{x}_N$ with associated features $(\mathbf{H}_n, \mathbf{p}_{\text{norm,n}}, s_n, m_n)$, $\forall n \in [1, ..., N]$. We designate $\mathbf{x}_0$ to be the target utterance for performing conversion. Then, the SSC loss $\mathcal{L}_{\text{ssc}}$ is defined as follows:

$$\hat{\mathbf{x}}_{n,0} = G(\mathbf{z}, \mathbf{H}_n, \mathbf{p}_{\text{norm},n}, s_0, m_0), \tag{4.9}$$

$$\mathcal{L}_{\text{ssc}} = \frac{1}{N} \sum_{n=1}^{N} \cos\big(E_s(\hat{\mathbf{x}}_{n,0}), s_0\big), \tag{4.10}$$

where $\cos(x_1, x_2)$ denotes the cosine similarity between x_1 and x_2.

4.3.3 GAN-based training

The generator and discriminator losses for training follow the least-squares GAN objective functions [55]. The overall losses are defined as follows:

$$\begin{aligned}\mathcal{L}_G = &\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}_{\mathbf{z},\mathbf{c}}\left[\left(D_k(G(\mathbf{z}, \mathbf{H}', \mathbf{p}_{\text{norm}}, s, m)) - 1\right)^2\right] \\ &+ \lambda_{\text{aux}}\mathcal{L}_{\text{aux}}(\mathbf{x}, G(\mathbf{z}, \mathbf{H}', \mathbf{p}_{\text{norm}}, s, m)) \\ &+ \lambda_{\text{ssc}}\mathcal{L}_{\text{ssc}},\end{aligned} \tag{4.11}$$

$$\mathcal{L}_D = \frac{1}{K}\sum_{k=1}^{K}\left(\mathbb{E}_{\mathbf{x}}\left[(D_k(\mathbf{x}) - 1)^2\right] + \mathbb{E}_{\mathbf{z},\mathbf{c}}\left[D_k(G(\mathbf{z}, \mathbf{H}', \mathbf{p}_{\text{norm}}, s, m))^2\right]\right), \tag{4.12}$$

where K denotes the number of all sub-discriminators across the MRSD and MPWD and D_k denotes the k-th sub-discriminator across all sub-discriminators. λ_{aux} and λ_{ssc} are weighting factors that balance the contributions of the auxiliary loss and SSC loss for the generator, respectively.

4.3.4 Training specifications

Throughout all of our experiments, we use audio sampled at a rate of 16 kHz. To obtain $\mathbf{H}$ from a time domain utterance $\mathbf{x}$, we start off with an 80-dimensional log-mel spectrogram $\mathbf{X}$ computed using a 1024 point Fourier transform, with a Hann window of 1024 samples and hop length of 256 samples. We then take the 20 lowest quefrency coefficients for low-quefrency liftering. To compute $\mathbf{H}'$, we choose the warping factor along the frequency axis from a uniform distribution over the range $[0.85, 1.15]$ for each training sample. $\mathbf{H}'$ thus has dimensions $(80, N)$, where N is the number of frames in the spectrogram $\mathbf{X}$. It is stacked with the other content feature, $\mathbf{p}_{\text{norm}}$ (which has dimensions $(257, N)$), and then fed into the model. Likewise, the two speaker features s and m are concatenated to produce a 576 ($= 512 + 64$) dimensional vector before being fed into the model.

For the speaker embeddings, we use a training strategy that is meant to make the model robust against small variations in the embedding values. For each speaker in the training set, we fit a GMM with 1 component[2] to the embeddings extracted from that speaker's training utterances. To reconstruct a given utterance during training, we randomly sample from the speaker's GMM to obtain the specific embedding s that is used for reconstruction. This ensures that a similar, but different speaker embedding is used to perform reconstruction for an utterance every time. We find that this strategy also helps the model generalize better to unseen speakers.

Training was done on four NVIDIA GeForce GTX 1080 Ti GPUs. We used the AdamW optimizer [53] with a learning rate of 1e-4 and $\beta_1 = 0.5, \beta_2 = 0.9$. All utterances and input features were cropped or padded to correspond to 16,384 samples (1 second) for batch processing. For the SSC loss, we set $N = 8$. Following [35], we set $\lambda_{\text{aux}} = 2.5$. Through empirical experiments, we set $\lambda_{\text{ssc}} = 0.9$.

The model was first trained with only self-reconstructive loss (without SSC loss) using a batch size of 32 for 1.8 million iterations. Then, we halved the learning rate to 5e-5 and continued training the model with the SSC loss included for 5,000 more iterations, using a decreased batch size of 16 due to GPU memory constraints. We found that this strategy ensured that the model learned to produce high-quality audio from the input features first, before then being guided to perform better voice conversion without compromising audio quality significantly. λ_{ssc} was linearly annealed from 0 to its final value for the first 2,000 steps in which the SSC loss was used.

4.4 Inference

Once training is complete, LVC-VC performs voice conversion at inference time by simply combining the content features from the source utterance and the speaker features from the target utterance to generate audio. Given source utterance $\mathbf{x}_1$ with content features $(\mathbf{H}_1, \mathbf{p}_{\text{norm},1})$ and target utterance $\mathbf{x}_2$ with speaker features (s_2, m_2),

[2] We assume that the distribution of speaker embeddings for a given speaker is roughly Gaussian with a single component.

the converted utterance $\hat{\mathbf{x}}_{1\rightarrow 2}$ is produced by:

$$\hat{\mathbf{x}}_{1\rightarrow 2} = G(\mathbf{z}, \mathbf{H}_1, \mathbf{p}_{\mathrm{norm},1}, s_2, m_2). \tag{4.13}$$

Chapter 5

From Voice Conversion to Anonymization

Until now, we have discussed our work largely in the context of voice conversion, where a source utterance is transformed to sound like a target speaker. However, recall that our original motivation is to perform voice *anonymization*. Although anonymization can be performed by using VC to change the perceived identity of a speaker to another individual, this requires that a specific target speaker—a real person—be chosen.

This brings up a variety of potential issues. First, target speakers must give permission for their voices to be used for the purposes of VC-based anonymization. In addition, there likely needs to be a reasonably large number of speakers in the pool of possible target voices in order to have enough options to satisfactorily anonymize a wide variety of source speakers. Finally, there is a major ethical issue regarding the potential of VC technologies to impersonate individuals—so-called "audio deepfakes". Given this, a VC-based anonymization methodology that could synthesize speech in a rich variety of non-existent speakers' voices would be an attractive prospect.

Inspired by the recently introduced task of *speaker generation* [98], this chapter introduces a methodology to use VC models for speaker anonymization without the need for specifying a target speaker. Although we discuss this approach in the context of LVC-VC, it can feasibly be used to extend the capabilities of any VC model that incorporates a speaker encoder.

5.1 Sampling Arbitrary Speaker Embeddings

At a high level, our idea for performing un-targeted speaker anonymization is straightforward: it involves modeling the distribution of speaker embeddings generated by the speaker encoder of a VC model and then sampling from that distribution to obtain an arbitrary speaker embedding. That embedding is then used as the "target" speaker embedding for the VC model in order to change the vocal characteristics of a given source utterance.

To do this, we use the speaker encoder E_s of LVC-VC to extract embeddings from a large number of speakers. Specifically, we do this for the speakers in the development set of the VoxCeleb1 dataset [65]; we randomly sample up to 50 utterances from each speaker, for a total of 60,402 embeddings from the 1,211 speakers in the dataset. Then, we fit a Gaussian mixture model (GMM) with 8 mixture components to the extracted embeddings; we refer to this GMM as S. To perform anonymization, we sample from S to extract an arbitrary speaker embedding $\tilde{s}$, which can then be fed into LVC-VC to transform and anonymize a source utterance.

5.2 Selecting F0

Recall that in addition to the speaker embedding, one of the speaker-related features that is fed into LVC-VC is a one-hot quantized representation of the target speaker's log F0. Therefore, randomly sampling a speaker embedding is not sufficient on its own to perform anonymization using LVC-VC; we must also specify the target voice's median F0 value in order to transform the source utterance.

It is possible to select a target F0 by randomly choosing a value from the range that is quantized: 65.4 Hz to 523.3 Hz (see Section 4.1.2). However, this could result in an F0 that is very different from the expected voice that the speaker embedding encodes. For example, the speaker embedding might correspond to a male-sounding voice, but the selected F0 value could be very high, corresponding to a female-sounding voice. We found during experiments that this mismatch could result in anonymized speech that sounded noisy, buzzy, or otherwise unnatural compared to the output of performing conversion on a specified target speaker.

To solve this issue, we trained a model F to predict the F0 of a voice from its corresponding speaker embedding $s \in \mathbb{R}^{512}$. The model is a feedforward neural network with one hidden layer of 512 units using ReLU activation and an output layer with 1 unit using sigmoid activation. Dropout [97] is used with $p = 0.5$. F is trained to predict the raw median F0 value of the voice corresponding to a speaker embedding. Specifically, it predicts a value between the minimum and maximum frequencies that are quantized by m (65.4 Hz ('C2') and 523.3 Hz ('C5')), which are normalized to be in the range $[0, 1]$.

We trained our F0 predictor model on the speaker embeddings of 40,017 utterances from 99 speakers in the VCTK corpus [120] and tested on the 4,225 utterances from the remaining 10 speakers. This was the same train-test split as we used for training LVC-VC and other baseline voice conversion models (see Section 6.1). We used the AdamW optimizer [53] with a learning rate of 1e-4 and $\beta_1 = 0.9, \beta_2 = 0.999$. On utterances from the test set, the model achieved a mean absolute F0 error of 12.43 Hz, with a standard deviation of 10.58 Hz. We determined that this level of performance was satisfactory for our purposes, since the objective of the F0 predictor was simply to approximate F0 values that would somewhat match random speaker embeddings that are sampled for performing anonymization.

5.3 Inference

Therefore, to perform un-targeted speaker anonymization with LVC-VC, we first sample an embedding $\tilde{s}$ from S. Then, we use the F0 predictor network F to estimate the median F0 of the voice that would correspond to $\tilde{s}$ and convert it into the one-hot quantized feature $\tilde{m}$. Given a source utterance $\mathbf{x}_1$ with associated content features $(\mathbf{H}_1, \mathbf{p}_{\text{norm},1})$, an anonymized version of that utterance $\tilde{\mathbf{x}}_1$ can be generated as follows:

$$\tilde{\mathbf{x}}_1 = G(\mathbf{z}, \mathbf{H}_1, \mathbf{p}_{\text{norm},1}, \tilde{s}, \tilde{m}). \tag{5.1}$$

Part III

Evaluation

Chapter 6

Data Sources

6.1 VCTK Corpus

We used the VCTK corpus [120] for model training and targeted voice conversion evaluation. The dataset consists of 44,242 utterances from 109 speakers, totaling around 44 hours of audio. Of the 109 speakers, 47 are male and 62 are female. The original audio has a sampling rate of 48 kHz, but we resampled it to 16 kHz for our experiments.

We randomly partitioned the dataset into 99 "seen" speakers, who would be used for many-to-many VC, and 10 "unseen" speakers, who would be used for zero-shot VC. The utterances of the 99 seen speakers were further randomly split into train and test sets in a 9-1 ratio. Only utterances from the seen speakers' train set were used for training models.

6.2 Local Voices Network (LVN)

We also used audio from Local Voices Network (LVN) conversations to evaluate the performance of LVC-VC; it was not used for training. As described in Section 1.1.1, LVN conversations usually have between 4 to 8 speakers and range from around 40 to 90 minutes in length. The lengths of speaker turns vary widely, from short bursts of frequent changes to long stretches of several minutes of just one person speaking. Conversations can be recorded either using a physical recording device or over a videoconferencing platform such as Zoom. Because of this, the audio quality of utterances can vary widely, with differing sources and amounts of background noise and reverberations depending on the acoustic environment or microphone setup of the speaker.

As of the writing of this thesis, the LVN platform contains almost 2,000 conversation recordings. However, conversations have privacy settings that limit how publicly visible they are. There are three levels of privacy:

- **Public:** Conversation is publicly available to anyone on the Internet.
- **Community:** Conversation is available to anyone with LVN account credentials.
- **Private:** Conversation is only available to a set of designated individuals.

We worked only with public conversations in order to avoid running into issues with data availability agreements. Furthermore, we filtered out all conversations that were not held in English (LVN also includes conversations that were held in Spanish). In total, there were 203 public conversations that met these criteria.[1] We preprocessed all audio by resampling to 16 kHz and splitting it up into chunks corresponding to individual sentences that each person spoke, according to the transcription and speaker diarization provided on the LVN platform. Then, we extracted 2 utterances per speaker from 50 speakers who were randomly sampled from all participants in these conversations, for a total of 100 utterances. These utterances ranged from 3 to 28 seconds in length, and were used as the source utterances in order to evaluate the performance of LVC-VC in terms of both targeted voice conversion and un-targeted voice anonymization tasks.

[1] As of April 2022.

Chapter 7

Evaluation Metrics

In this chapter, we outline the various metrics that we use to evaluate models on voice conversion and anonymization tasks. These take the form of objective and subjective measurements that aim to quantify a model's performance in terms of generating properly converted or anonymized audio. In particular, we evaluated three aspects of the generated speech:

- **Quality/Naturalness:** How free of noise or other artifacts is the converted or anonymized speech? How natural does the audio sound?

- **Intelligibility:** How well does the converted or anonymized speech preserve the linguistic information of the original source utterance? In other words, how well is the pronunciation of words maintained?

- **Similarity:** For voice conversion, how similar does the transformed utterance sound to the target speaker? For anonymization, how dissimilar does the transformed utterance sound to the source speaker?

Although we did not use exactly the same methodologies, our evaluation metrics and protocols were inspired in part by the evaluation plan of the VoicePrivacy Challenge [108].

7.1 Objective Metrics

7.1.1 Quality

Although developing an objective measure to evaluate the general quality of speech is difficult, several methods have been introduced in recent years to automatically estimate speech quality. These methods are meant to act as proxy measures to subjective evaluations of audio quality done through human listening tests, which can be costly and time-consuming. For our purposes, we used NISQA [61] to compute objective quality scores for how clean utterances sounded. NISQA is trained to evaluate the overall quality of an audio sample, taking into account four dimensions: noisiness, coloration, discontinuity, and loudness. Specifically, it estimates the mean opinion score (MOS) for overall speech quality that a human would assign to the utterance. Scores are given in the range 1–5, where 1 means that the audio quality is very poor and 5 means that the audio quality is very clean.

7.1.2 Intelligibility

For an objective measure of an utterance's intelligibility, we computed its word error rate (WER) and character error rate (CER) on an automatic speech recognition (ASR) task against the ground truth transcript. We used a pre-trained wav2vec 2.0 model [5] made available through the Hugging Face Transformers library [117]. Specifically, we used the "wav2vec2-base-960h" version of the model, which was pretrained and fine-tuned to perform ASR on 960 hours of the LibriSpeech corpus [75].

7.1.3 Speaker similarity

To objectively evaluate speaker similarity, we performed an automatic speaker verification (ASV) task on the converted or anonymized utterances and computed the equal error rate (EER), which is the threshold at which the false acceptance and false rejection rates of the verification system are equal. For fair evaluation, we used a different speaker verification model from the speaker encoder that is used in LVC-VC (see Section 4.2.2). Specifically, we used a slightly larger, more performance-optimized ResNet-34-based model from [29]. This model contains half the number of channels in each residual block compared to the original ResNet-34 (around 8.0 million parameters) and uses attentive statistics pooling [70] to aggregate temporal frames. It was trained on the VoxCeleb2 development set and achieves an EER of 1.18% on the VoxCeleb1 test set.

For targeted voice conversion, we compared a converted utterance against a different randomly sampled utterance from the source speaker as well as against the target utterance; the goal was to see whether the converted utterance would be identified as a different speaker from the source speaker and the same speaker as the target speaker. For un-targeted anonymization, we simply compared a transformed utterance against a different randomly sampled utterance from the source speaker in order to see if the identity had been successfully anonymized.

7.2 Subjective Metrics

All subjective listening tests were conducted on Amazon Mechanical Turk. For each of the listening tests described below, each utterance or utterance pair was evaluated by two subjects. Further details on the full survey design and data collection procedures are described in Appendix A.

7.2.1 Naturalness

We conducted mean opinion score (MOS) tests to evaluate the naturalness of utterances. Note that the naturalness score here is somewhat different from the objective quality metric described above in Section 7.1.1, as humans take into account the vocal characteristics of a speech sample as well as its overall cleanliness to determine naturalness. Here, subjects were asked to assign a score from 1–5 on the naturalness of each utterance, where 1 meant that the utterance did not sound natural at all and 5 meant that the utterance sounded completely natural.

7.2.2 Intelligibility

We also conducted tests to evaluate the intelligibility of utterances. For each utterance, subjects were asked to assign a score from 1–5 on the intelligibility of the audio, where 1 meant that the utterance was not understandable at all and 5 meant that the utterance was perfectly understandable. Listeners were instructed to ignore the general audio quality of the utterance and focus explicitly on whether the pronunciation of the spoken words was clear or not.

7.2.3 Speaker similarity

Judging the similarity of one speaker to another is a rather unusual task that is not an element of peoples' everyday speech perception. However, recognizing speakers is something that is done all the time. Therefore, we followed the methodology described in [115] and designed our subjective speaker similarity evaluation to be more like that of a speaker recognition task. Speakers were presented with pairs of utterances and asked to indicate whether the two voices sounded like they came from the same speaker. The scale for judging was: "Same speaker: Absolutely sure", "Same speaker: Not sure", "Different speaker: Not sure", and "Different speaker: Absolutely sure". Then, the responses were converted to binary decisions of "same" or "different", after which we measured the percentage of each response for a given model.

For voice conversion, each pair of utterances consisted of a converted utterance and the corresponding target utterance. For anonymization, each pair consisted of an anonymized utterance and a different random utterance spoken by the source speaker.

Chapter 8

Results

In this chapter, we present the results from using LVC-VC to perform voice conversion and anonymization on the VCTK and LVN datasets described in Chapter 6, as well as performance comparisons against other current state-of-the-art voice conversion models.

8.1 VCTK

8.1.1 Targeted voice conversion

To evaluate the performance of LVC-VC on targeted voice conversion, we compared it against six other VC models: AdaIN-VC [10], AGAIN-VC [8], AutoVC [86], AutoVC-F0 [83], Blow [91], and NVC-Net [68]. AdaIN-VC, AGAIN-VC, AutoVC, and AutoVC-F0 are not end-to-end models; they produce spectrograms, which must then be passed through a vocoder to produce time domain audio. Blow and NVC-Net are end-to-end models; they take audio waveforms as input and directly produce time domain audio.

We used the official implementations of all of the models from GitHub except for AutoVC-F0, which we implemented according to the instructions in the paper since an official implementation was not publicly available. All models were trained from scratch on the same train-test split of the VCTK dataset as LVC-VC. For a fair comparison, models taking spectrograms as input features were trained using the same spectrogram configuration as LVC-VC, and all time-domain audio was synthesized using a UnivNet-c16 vocoder [35] that was trained on the LibriTTS dataset [125].

We considered three different voice conversion settings for evaluation:

- Seen-to-seen: Conversion from a "seen" speaker in the training set to another "seen" speaker.
- Unseen-to-seen: Conversion from an "unseen" speaker not in the training set to a "seen" speaker in the training set.
- Unseen-to-seen: Conversion from an "unseen" speaker not in the training set to another "unseen" speaker; i.e. true zero-shot voice conversion.

All of the VC models we used for comparison are capable of performing zero-shot voice conversion except for Blow, which can only convert voices to and from previously seen speakers. Therefore, we only evaluated Blow on the seen-to-seen setting, while all other models were evaluated on all three settings.

For every voice conversion setting, we considered four gender-to-gender conversion combinations: male-to-male, male-to-female, female-to-male, and female-to-female. For the seen-to-seen setting, we sampled 25 speakers from the 99 seen speakers in the VCTK corpus as source speakers and randomly assigned one speaker from each gender to act as target speakers. For each source-target speaker pair, we randomly sampled two utterances from each speaker for performing conversion. This resulted in 200 ($= 25 \times 2 \times 4$) utterance pairs for seen-to-seen conversion. For the unseen-to-seen setting, we used the 10 unseen speakers from the VCTK corpus as source speakers and randomly sampled target speakers and utterances from the 99 seen speakers in the same way as above. This resulted in 80 ($= 10 \times 2 \times 4$) utterance pairs. Finally, for the unseen-to-unseen setting, we used the 10 unseen speakers from the VCTK corpus as source speakers and randomly sampled target speakers and utterances from the other 9 unseen speakers, resulting in 80 ($= 10 \times 2 \times 4$) utterance pairs.

Tables 8.1, 8.2, and 8.3 show the results of evaluating the aforementioned voice conversion models with the metrics described in Chapter 7 on the seen-to-seen,

Table 8.1: Seen-to-seen voice conversion evaluation results on the VCTK dataset. (MOS: Mean opinion score for naturalness (1–5); INT: Mean subjective intelligibility score (1–5); SIM: Subjective similarity score of utterances (%); WER: ASR word error rate (%); CER: ASR character error rate (%); EER: ASV equal error rate (%); NISQA: Average quality score predicted by NISQA model.)

Model	**MOS**	**INT**	**SIM**	**WER**	**CER**	**EER**	**NISQA**
Ground Truth	4.40 ± 0.07	4.64 ± 0.06	91.75	11.27	3.94	1.50	4.50 ± 0.05
UnivNet (Vocoder)	4.33 ± 0.08	4.47 ± 0.07	90.50	12.26	4.47	2.00	4.45 ± 0.06
LVC-VC	3.54 ± 0.12	4.17 ± 0.10	46.00	22.69	9.55	18.50	4.00 ± 0.08
AdaIN-VC	2.33 ± 0.10	3.05 ± 0.13	53.00	43.06	22.48	33.00	3.75 ± 0.09
AGAIN-VC	2.04 ± 0.10	2.88 ± 0.13	44.00	47.64	25.22	24.00	3.70 ± 0.10
AutoVC	3.78 ± 0.10	4.15 ± 0.09	22.25	24.24	10.82	40.50	4.13 ± 0.06
AutoVC-F0	3.59 ± 0.10	4.03 ± 0.10	34.50	25.16	11.81	33.00	4.10 ± 0.07
Blow	1.85 ± 0.08	3.19 ± 0.13	35.00	31.01	14.86	53.00	3.07 ± 0.11
NVC-Net	2.96 ± 0.11	3.40 ± 0.13	67.75	48.91	27.25	15.00	4.31 ± 0.07

Table 8.2: Unseen-to-seen voice conversion evaluation results on the VCTK dataset.

Model	**MOS**	**INT**	**SIM**	**WER**	**CER**	**EER**	**NISQA**
Ground Truth	4.43 ± 0.12	4.83 ± 0.07	93.75	9.69	2.93	0.00	4.42 ± 0.09
UnivNet (Vocoder)	4.34 ± 0.11	4.55 ± 0.12	93.75	11.70	3.77	0.00	4.38 ± 0.10
LVC-VC	3.31 ± 0.15	4.31 ± 0.14	41.88	17.37	7.03	20.00	3.89 ± 0.14
AdaIN-VC	2.42 ± 0.15	3.23 ± 0.18	53.75	36.56	17.86	36.25	3.85 ± 0.13
AGAIN-VC	2.43 ± 0.15	3.34 ± 0.20	45.63	43.33	21.62	33.75	3.72 ± 0.18
AutoVC	3.50 ± 0.13	4.33 ± 0.14	28.75	23.03	10.97	30.00	4.18 ± 0.11
AutoVC-F0	3.52 ± 0.14	4.06 ± 0.16	38.13	22.12	9.67	26.25	4.04 ± 0.12
NVC-Net	3.17 ± 0.18	3.44 ± 0.21	60.63	48.45	26.91	11.25	4.14 ± 0.13

Table 8.3: Unseen-to-unseen voice conversion evaluation results on the VCTK dataset.

Model	**MOS**	**INT**	**SIM**	**WER**	**CER**	**EER**	**NISQA**
Ground Truth	4.41 ± 0.12	4.73 ± 0.08	93.75	12.06	3.58	0.00	4.37 ± 0.10
UnivNet (Vocoder)	4.36 ± 0.10	4.67 ± 0.09	91.25	14.34	4.85	0.00	4.37 ± 0.09
LVC-VC	3.08 ± 0.14	4.06 ± 0.16	29.38	20.10	8.29	26.25	3.50 ± 0.13
AdaIN-VC	2.41 ± 0.14	3.28 ± 0.21	50.63	41.43	20.53	35.00	3.55 ± 0.17
AGAIN-VC	2.18 ± 0.14	2.90 ± 0.20	30.00	51.57	26.86	32.50	3.35 ± 0.16
AutoVC	3.39 ± 0.16	4.00 ± 0.16	5.63	27.97	12.41	66.25	4.09 ± 0.10
AutoVC-F0	3.21 ± 0.15	4.08 ± 0.16	12.50	28.15	12.55	63.75	3.94 ± 0.12
NVC-Net	3.09 ± 0.16	3.44 ± 0.20	35.63	50.51	26.27	37.50	4.24 ± 0.11

unseen-to-seen, and unseen-to-unseen settings, respectively. We also report the results for ground truth speech and speech reconstructed using the UnivNet vocoder to provide baseline values for reference. MOS denotes the subjective naturalness score (Section 7.2.1), INT denotes the subjective intelligibility score (Section 7.2.2), and SIM denotes the subjective similarity score (Section 7.2.3). WER and CER are the ASR word error and character error rates (Section 7.1.2), EER is the ASV equal error rate (Section 7.1.3), and NISQA is the audio quality score estimated by the NISQA model (Section 7.1.1). For MOS, INT, SIM, and NISQA, higher is better. For WER, CER, and EER, lower is better. We report average scores across all converted utterances for MOS, INT, WER, CER, and NISQA. We also report 95% confidence intervals for MOS, INT, and NISQA.

We see that most of the previously proposed voice conversion models largely fall under two categories: 1) those that are able to perform voice style transfer (VST) reasonably well, but produce low-quality or less intelligible audio (AdaIN-VC, AGAIN-VC, NVC-Net), and 2) those that produce high-quality and intelligible audio, but are not able to perform VST very well (AutoVC, AutoVC-F0). In other words, all of these models appear to face a trade-off between producing high-quality audio and achieving good VST performance.

AdaIN-VC and AGAIN-VC are able to produce audio with relatively good scores for SIM and EER, but they do poorly in terms of MOS, NISQA, INT, WER, and CER. NVC-Net actually appears to produce the cleanest audio—it achieves the best NISQA scores in all three settings—and it performs quite well in terms of VST, achieving the best SIM and EER in the seen-to-seen and unseen-to-seen settings. However, it has among the worst WER and CER scores and a fairly low INT score, indicating that it is not able to preserve content information very well; this appears to contribute to its lower MOS. In addition, its VST performance degrades significantly in the unseen-to-unseen setting.

AutoVC and AutoVC-F0 demonstrate relatively good scores for MOS, INT, WER, CER, and NISQA in all three settings, indicating that they are able to produce clean, intelligible audio. However, they perform quite poorly in terms of VST, as shown by their SIM and EER scores. This is especially evident in the unseen-to-unseen setting, where the SIM and EER scores are by far the worst among all of the models.

LVC-VC is able to manage these trade-offs much better than the other models. While it is not quite the best at producing clean, natural audio, it achieves MOS and NISQA scores that are competitive with the other best models in those categories, especially in the seen-to-seen and unseen-to-seen settings. Notably, it achieves very high INT scores and has by far the lowest WER and CER in all three settings, indicating that it is able to maintain the linguistic content and pronunciation clarity of source utterances very well. This shows that the low-quefrency liftered mel-spectrogram that we use as LVC-VC's input feature is effective at preserving and passing on the content information of source utterances to converted utterances. LVC-VC also performs quite well in terms of VST performance. Although it does not quite outperform AdaIN-VC and AGAIN-VC in terms of SIM, it obtains a better EER than them in all three settings. It also obtains the best EER among all of the models in the unseen-to-unseen setting.

8.1.2 Un-targeted voice anonymization

Following the methodology outlined in Chapter 5, we performed un-targeted voice anonymization on the same 80 utterances that were used for evaluating unseen-to-seen targeted voice conversion on the VCTK corpus. Here, because the goal is to perform anonymization, lower SIM and higher EER scores are better; 0.00% SIM and 50.00% EER would indicate perfect anonymization for all utterances. Table 8.4 shows the results of performing anonymization in this way.

We find that our anonymization method appears to work quite well for masking speaker identity. SIM decreased from 93.75% to 26.88% and EER increased from 0.00% to 31.25%, indicating that speakers' voices were successfully masked in most cases for both human listeners and the ASV model. We note that we did not put any particular constraints when sampling speaker embeddings for anonymization (e.g. by enforcing that the sampled embeddings have a minimum cosine distance from the source utterance's embedding); adding these constraints would likely result in an even

Table 8.4: Un-targeted voice anonymization evaluation results on the VCTK dataset.

Model	**MOS**	**INT**	**SIM**	**WER**	**CER**	**EER**	**NISQA**
Ground Truth	4.43 ± 0.12	4.83 ± 0.07	93.75	9.69	2.93	0.00	4.42 ± 0.09
LVC-VC: Untargeted	3.16 ± 0.17	3.70 ± 0.16	26.88	15.17	6.05	31.25	3.70 ± 0.14

greater degree of anonymization on average.

As a side effect of anonymization, however, we do see that the naturalness, intelligibility, and general audio quality of anonymized speech decreases compared to the original ground truth audio (although WER and CER are not as adversely affected). This is likely a result of LVC-VC being fed speaker embeddings that have been sampled from a latent space of embeddings that it has not seen before. We hypothesize that this quality degradation could be mitigated to some extent if LVC-VC were trained on a larger dataset with many more speakers; being exposed to a wider space of speaker embeddings during training could help the model generalize better to arbitrary speaker embeddings during inference.

In practice, we believe it would be possible to design an un-targeted speaker anonymization system by randomly sampling several different potential target embeddings, and then allowing people who wish to anonymize their voices to listen to the options and select the target voice that they would like to use. By giving human users some degree of control over the overall process, it should be possible to guarantee a satisfactory level of voice anonymization while better maintaining the perceptual elements of the transformed speech.

8.1.3 Ablation studies

We conducted ablation studies on LVC-VC to evaluate the impact of the various input features and training strategies on the model's performance. Specifically, we tested versions of the model trained without sampling speaker embeddings for reconstructive training from GMMs (using a single average embedding for each speaker instead), without the SSC loss, without warping the low-quefrency liftered mel spectrogram $\mathbf{H}$, and without using each of the input features $\mathbf{p}_{\text{norm}}$ and m. The results on seen-

Table 8.5: Seen-to-seen voice conversion evaluation results for various ablations of LVC-VC on the VCTK dataset.

Model	**WER**	**CER**	**EER**	**NISQA**
LVC-VC	22.69	9.55	18.50	4.00 ± 0.08
w/o GMM embeddings	23.33	10.18	15.50	3.86 ± 0.09
w/o SSC loss	15.79	5.97	68.00	4.16 ± 0.08
w/o warping $\mathbf{H}$	19.80	8.51	41.50	3.96 ± 0.09
w/o $\mathbf{p}_{\text{norm}}$	23.11	10.22	18.50	3.71 ± 0.10
w/o m	22.34	9.05	21.00	3.91 ± 0.09

Table 8.6: Unseen-to-seen voice conversion evaluation results for various ablations of LVC-VC on the VCTK dataset.

Model	**WER**	**CER**	**EER**	**NISQA**
LVC-VC	17.37	7.03	20.00	3.89 ± 0.14
w/o GMM embeddings	16.64	7.10	16.25	3.69 ± 0.15
w/o SSC loss	12.25	4.38	71.25	4.04 ± 0.13
w/o warping $\mathbf{H}$	19.56	8.40	42.50	3.81 ± 0.17
w/o $\mathbf{p}_{\text{norm}}$	19.01	7.61	25.00	3.66 ± 0.15
w/o m	18.28	7.46	20.00	3.82 ± 0.13

Table 8.7: Unseen-to-unseen voice conversion evaluation results for various ablations of LVC-VC on the VCTK dataset.

Model	**WER**	**CER**	**EER**	**NISQA**
LVC-VC	20.10	8.29	26.25	3.50 ± 0.13
w/o GMM embeddings	26.92	11.11	25.00	2.89 ± 0.14
w/o SSC loss	16.78	6.64	68.75	3.83 ± 0.13
w/o warping $\mathbf{H}$	19.76	7.39	51.25	3.62 ± 0.17
w/o $\mathbf{p}_{\text{norm}}$	21.33	8.60	32.50	3.36 ± 0.18
w/o m	22.90	9.90	28.75	3.47 ± 0.14

to-seen, unseen-to-seen, and unseen-to-unseen voice conversion are shown in Tables 8.5, 8.6, and 8.7, respectively. For convenience, we only report scores from objective metrics.

When LVC-VC is trained on a single average speaker embedding instead of sampling from a GMM, we find that VST performance improves slightly. However, the overall audio quality decreases significantly, especially in the unseen-to-unseen setting. This suggests that training the model to reconstruct audio from diverse speaker embeddings sampled from a GMM helps it combine the information in speaker embeddings with content features more coherently when synthesizing audio. Consequently, the model is also able to better utilize speaker embeddings from unseen speakers, leading it to produce much higher quality audio in the zero-shot setting.

When we train LVC-VC without SSC loss, we find that the model has trouble performing VST accurately, as evidenced by much higher EERs in all three settings. This demonstrates the importance of explicitly guiding the model to perform voice conversion rather than only relying on self-reconstructive training. Warping the low-quefrency liftered mel spectrogram has a similar effect on VST performance; training LVC-VC using $\mathbf{H}$ rather than $\mathbf{H}'$ for self-reconstruction also leads to significantly higher EERs. This indicates that, without warping, the source speaker information remaining in $\mathbf{H}$ seems to leak through to the output audio and cause imperfect conversion. Meanwhile, training using the warped feature $\mathbf{H}'$ seems to effectively perturb the source speaker information such that LVC-VC is able to "un-warp" the content features to match the vocal characteristics of the target speaker much more accurately.

Finally, while they do not appear to crucially impact any one aspect of the model's performance, the normalized F0 contour $\mathbf{p}_{\text{norm}}$ and quantized F0 median m contribute to relatively small, but significant performance gains in terms of all measured metrics. Therefore, we can see that each of the training strategies and features that we used contribute meaningfully to the overall performance of LVC-VC.

8.2 LVN

We used LVC-VC to anonymize utterances that were sampled from LVN conversations as described in Section 6.2. We used two different methods: 1) targeted anonymization, where the utterances were converted to the voices of seen speakers from the VCTK corpus, and 2) un-targeted anonymization, where we followed the methodology described in Chapter 5. For targeted anonymization, we randomly selected a speaker from the 99 seen speakers in the VCTK corpus to use as the target speaker. As before, since we are performing anonymization, lower SIM and higher EER scores

Table 8.8: Targeted and un-targeted voice anonymization evaluations results on audio from LVN conversations.

Model	MOS	INT	SIM	WER	CER	EER	NISQA
Ground Truth	4.49 ± 0.10	4.50 ± 0.11	91.00	38.16	20.72	2.00	3.27 ± 0.17
LVC-VC: Targeted	2.21 ± 0.13	2.71 ± 0.15	25.00	65.98	36.86	31.00	2.91 ± 0.15
LVC-VC: Untargeted	2.02 ± 0.12	2.33 ± 0.14	10.50	69.61	38.97	39.00	2.55 ± 0.14

are better. The results are shown in Table 8.8.

Before diving into the results, it is worth noting several differences between the audio from LVN and VCTK. First, LVN audio quality is significantly worse than that of VCTK. Compared to NISQA scores around 4.4 or 4.5 for VCTK, LVN data only has an average NISQA score of 3.27. This is largely a result of the diverse conditions under which LVN conversations are recorded, which includes varying amounts of background noise and reverberation.

Second, the WER and CER for ground truth LVN audio is significantly worse than for VCTK. This may partly be due to the relatively poor audio quality, which could make it difficult for the wav2vec 2.0 ASR model to accurately recognize the content of the speech. However, it is likely also a consequence of inaccurate speech transcriptions. On the LVN platform, transcripts are meant to be read on their own or followed along while listening to the audio; consequently, many artifacts of natural human speech, such as "um"s, "uh"s, or repeated words or phrases, are not transcribed verbatim for the sake of readability. This phenomenon likely also contributes to the much higher ASR error rates that we see here.

In terms of the anonymization performance of LVC-VC, we see a similar pattern to the results seen above for VCTK. Both targeted and un-targeted anonymization methods result in low SIM scores and high EERs (25.00% and 10.50%, and 31.00% and 39.00%, respectively), indicating successful anonymization of vocal identity in most cases. However, we also see a fairly large degradation in speech quality, both in terms of naturalness and intelligibility. Subjective naturalness and intelligibility scores notably decrease quite significantly, and ASR error rates and NISQA scores also worsen as well.

Given the way in which LVC-VC was trained, these results are not entirely surprising. Because LVC-VC was trained exclusively on clean audio from VCTK, it was never exposed to noisy or reverberant audio. Therefore, it may not be able to properly utilize the low-quefrency liftered spectrograms of utterances with a poor signal-to-noise ratio, as the frequency characteristics of noise and reverberations likely muddle the content information that is encoded in the input feature. We discuss this issue more in-depth and propose some strategies for mitigating these shortcomings in Chapter 10.

Chapter 9

Internal Representations: What's going on under the hood?

We have seen that LVC-VC is able to synthesize audio by combining together various content and speaker-related input features within a vocoder-like framework. However, we do not know *how* exactly the audio is generated. What is actually happening inside the model? Is the intuition we described previously—that speaker features are used to "un-normalize" and "un-warp" the content features—correct? We performed several explorations of the intermediate representations of LVC-VC in order to better understand how the model works. This chapter describes the results of those analyses.

9.1 Time Domain Audio Generation

To investigate how LVC-VC generates time domain audio, we performed spectral analyses of the intermediate outputs of the model after each transposed convolutional block. Specifically, we looked at linear spectrograms of each of these intermediate outputs in order to gain an intuition of what is happening at each step as the model upsamples the input noise sequence to eventually produce the output audio signal.

Recall that LVC-VC starts with an input noise sequence that is at a $\frac{1}{256}\times$ temporal resolution compared to the final output signal. It contains three 1D transposed convolutional layers that upsample this input sequence by 8×, 8×, and 4× to produce

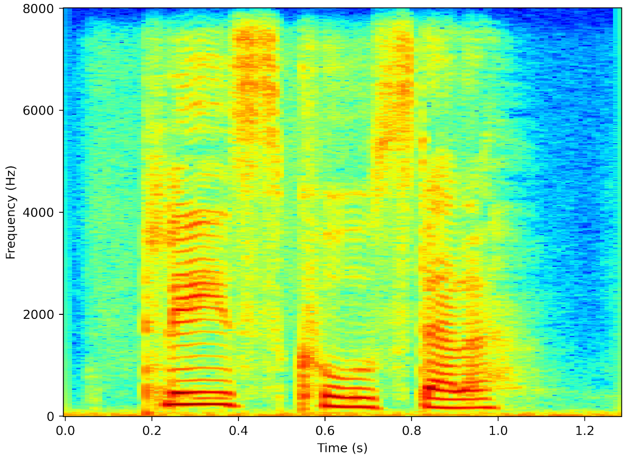


Figure 9-1: Spectrogram of the sample utterance that we use to illustrate spectral analyses of the internal representations of LVC-VC.

the final time domain waveform. Therefore, we can essentially consider the outputs of the three transposed convolutional blocks to be downsampled versions of the final output signal with temporal resolutions that are $\frac{1}{32}\times$, $\frac{1}{4}\times$, and $1\times$ that of real-time. Taking this into account, we computed the STFTs of these downsampled signals using the following Fourier transform parameters:

- $\frac{1}{32}\times$ downsampled signal: 32 point Fourier transform, 32 sample window length, 8 sample hop length, corresponding to audio sampled at 500 Hz.

- $\frac{1}{4}\times$ downsampled signal: 256 point Fourier transform, 256 sample window length, 64 sample hop length, corresponding to audio sampled at 4 kHz.

- $1\times$ downsampled signal: 1024 point Fourier transform, 1024 sample window length, 256 sample hop length, corresponding to audio sampled at 16 kHz.

Note that because the model uses 16 channels in its convolutional layers, the output of each transposed convolutional block also has 16 channels.

In the rest of this chapter, we use a sample utterance from the VCTK corpus to illustrate the results of the spectral analyses we performed on the internal representations of LVC-VC. The utterance is of the phrase “Please call Stella” and is spoken by a female voice. Figure 9-1 shows the linear spectrogram corresponding to this utterance.

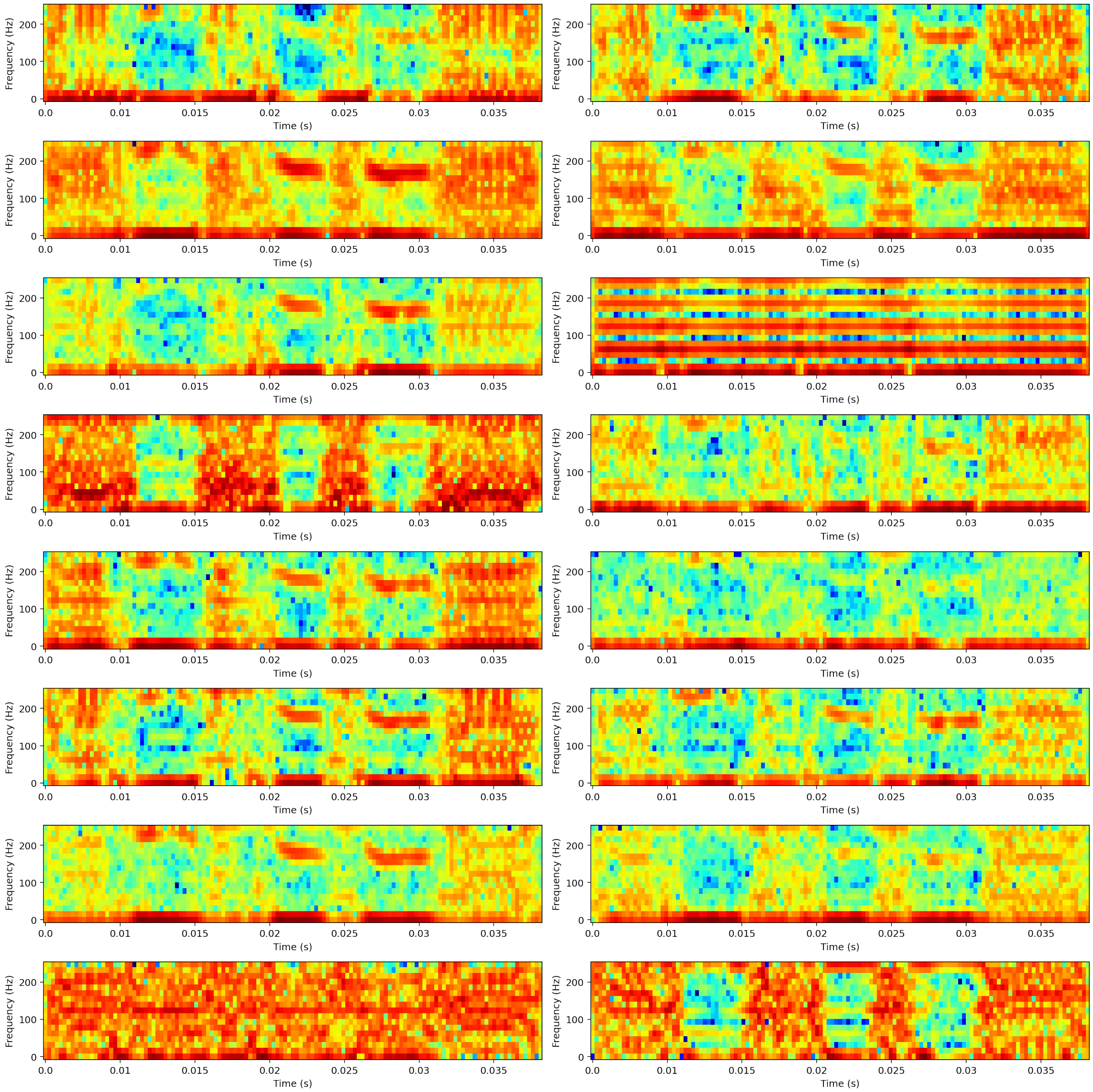


Figure 9-2: Results of computing the STFT on the output of the first transposed convolutional stack of LVC-VC. Signals are at a $\frac{1}{32}\times$ temporal resolution compared to real-time.

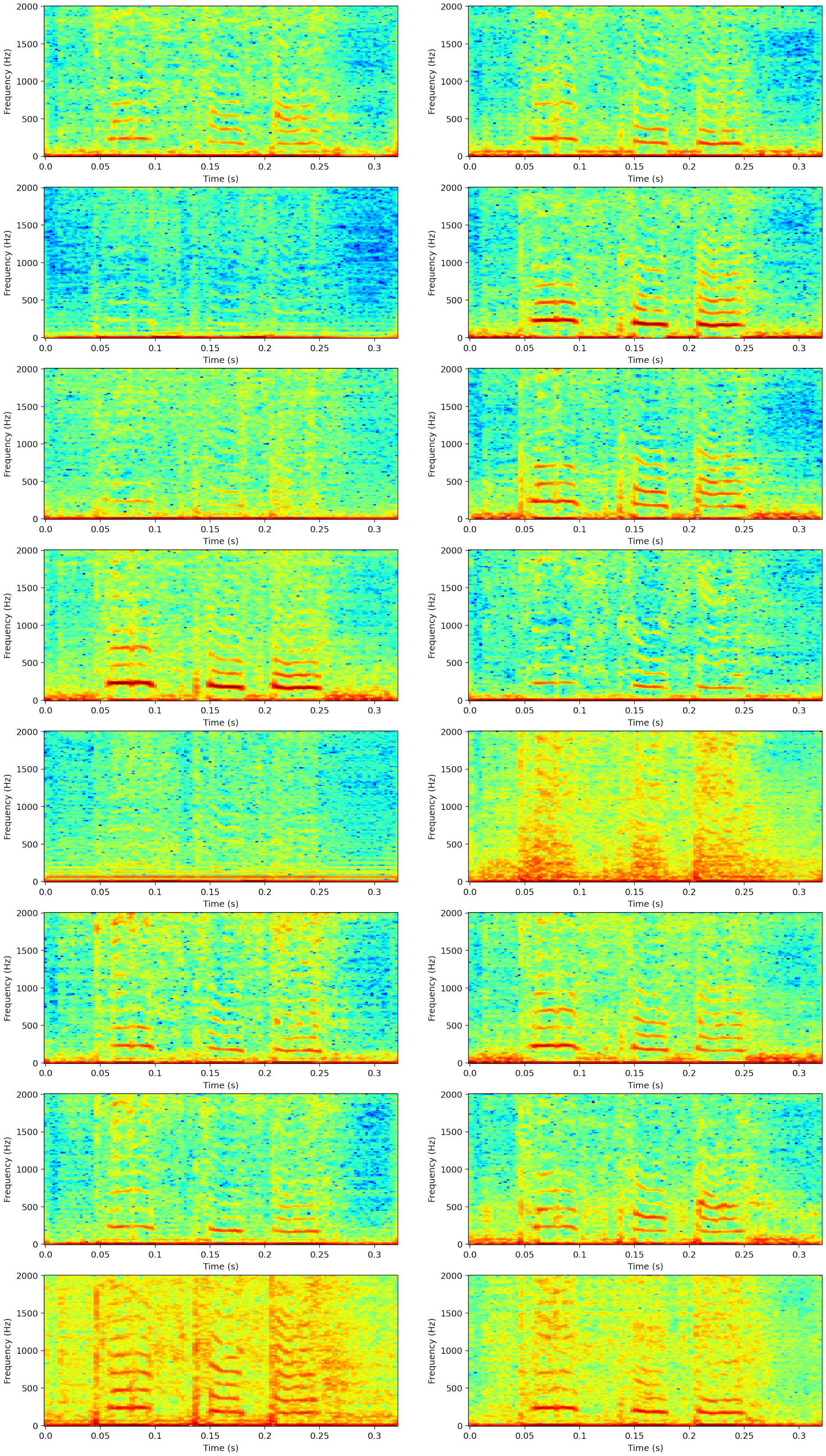


Figure 9-3: Results of computing the STFT on the output of the second transposed convolutional stack of LVC-VC. Signals are at a $\frac{1}{4}\times$ temporal resolution compared to real-time.

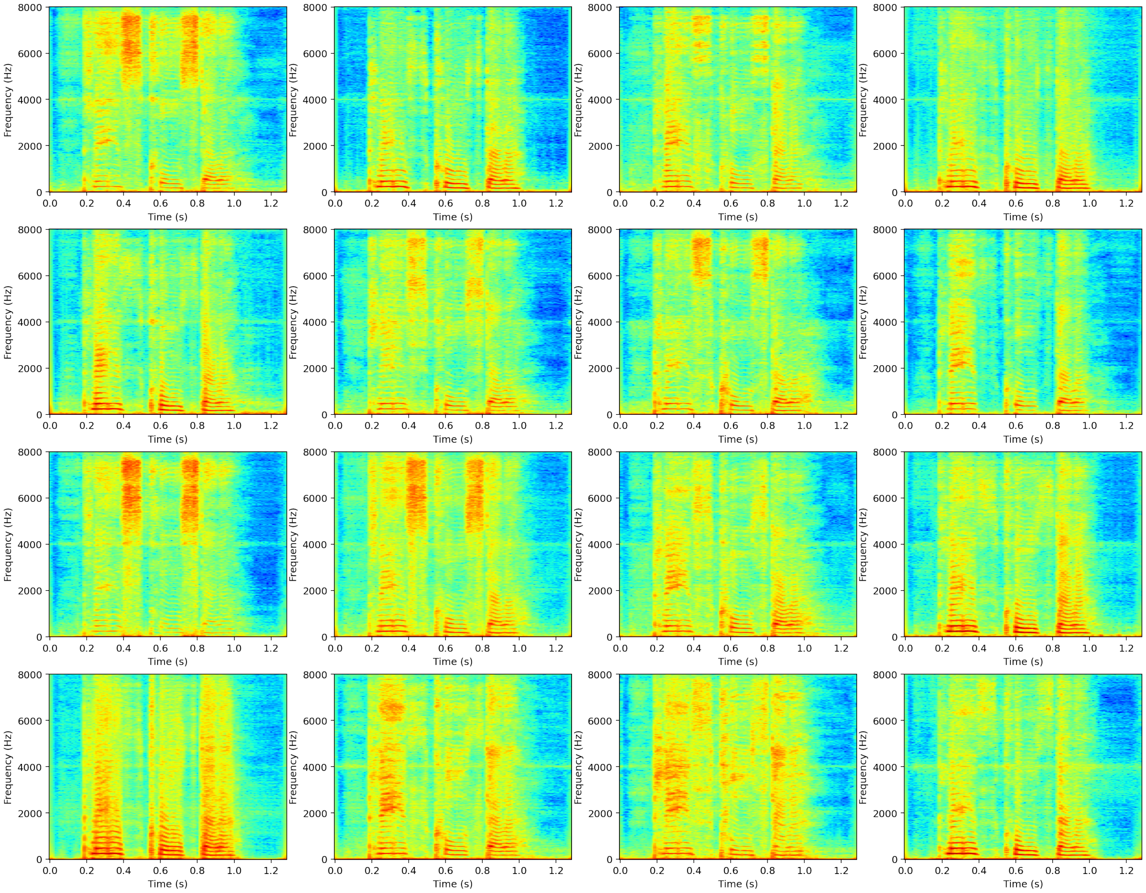


Figure 9-4: Results of computing the STFT on the output of the third transposed convolutional stack of LVC-VC. Signals are at the same temporal resolution (1×) as real-time.

Figure 9-2 shows the results of computing the STFT on the 16 channels of the output of the first transposed convolution stack when LVC-VC is tasked with reconstructing the sample utterance. We can see that the model immediately begins to construct content information, as demonstrated by the emergence of three clear voiced segments in the STFT outputs ("Please", "call", "Stella"). In these voiced segments, we also see the emergence of the first harmonic band of the speaker's F0 around the 200 Hz mark. In addition, we can notice that individual channels appear to model different aspects of the audio signal. Some channels appear to model the voiced segments of the utterance, while others appear to model the unvoiced segments, background noise, or silence.

Figures 9-3 and 9-4 show the corresponding STFT visualizations for the outputs of the second and third transposed convolution stacks, respectively. We see that similar patterns emerge in terms of different channels appearing to correspond to different aspects of the final time domain audio signal: voiced segments, unvoiced segments, background noise, and silence. In Figure 9-3, we can notably see one channel at (5, 2) that appears to encode the spectral envelope and the formant frequencies of the utterance over time. We also see the formation of more harmonic frequencies and the overall F0 contour, indicating the gradual addition of more detailed speaker and content information as the signal propagates through the model.

Overall, these results indicate that the different channels of LVC-VC's convolutional layers encode the various aspects of a speech signal, such as voiced and unvoiced segments, vocal cord harmonics, formants, silence, and noise. LVC-VC thus appears to generate audio by starting off with incorporating high-level speaker and content features in the lower layers of the model, and then gradually adding in more fine-grained speaker and content information as it dilates the signal across the time domain and spectrum. These results are perhaps not entirely surprising; we can see them as analogous to the way in which convolutional filters in deep computer vision models learn to encode different aspects of images in their layers, such as edges, colors, and patterns [48, 71].

9.2 Incorporation of Speaker and Content Information

Now that we have gained an intuition of how LVC-VC synthesizes audio overall, we would like to see how it incorporates speaker and content information during the audio generation process. To do this, we performed experiments where we ablated each of these features and performed spectral analyses of the resulting outputs.

9.2.1 Speaker information

To analyze how LVC-VC incorporates speaker information in the speech generation process, we zeroed out the speaker embedding s and quantized log median F0 m and made the model generate audio using only content features. Then, we performed spectral analyses in the same way as in Section 9.1 by computing the STFTs of the intermediate outputs of the transposed convolution stacks as well as of the final output signal. Figure 9-5 illustrates the final output of the model when it generates audio with no speaker information, and Figure 9-6 illustrates the intermediate outputs of the transposed convolution stacks. For brevity, we only include visualizations for 4 out of the 16 channels for each of the intermediate outputs.

We can see that the spectral envelope and formants of the utterance are well

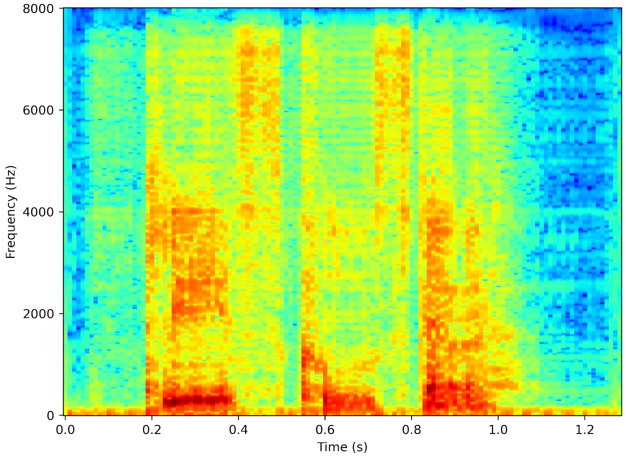
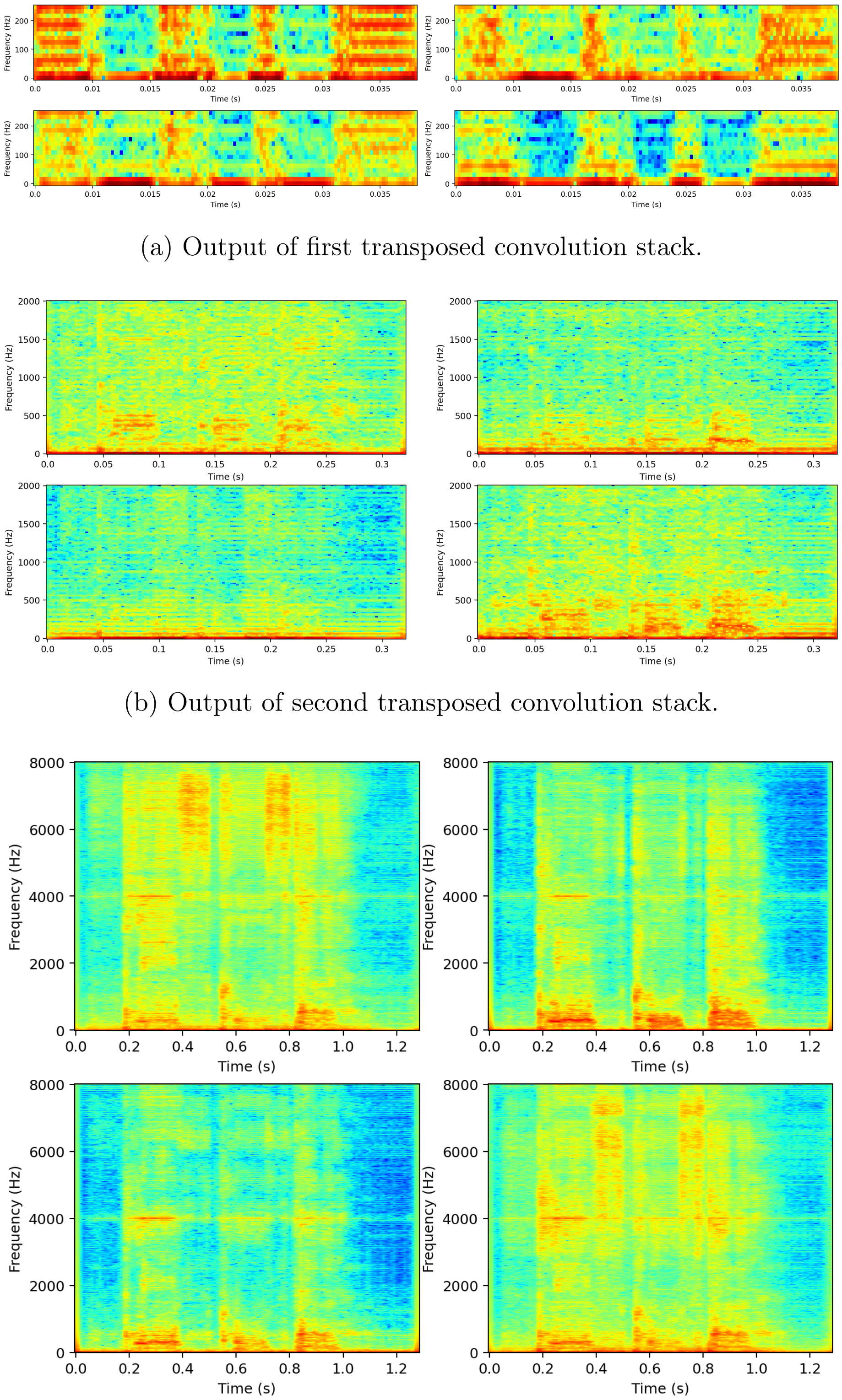


Figure 9-5: Spectrogram computed from the output signal generated by LVC-VC when the speaker embedding s and quantized log median F0 m have been zeroed out.



(a) Output of first transposed convolution stack.

(b) Output of second transposed convolution stack.

(c) Output of third transposed convolution stack.

Figure 9-6: Results of computing the STFT on the intermediate outputs of LVC-VC after each transposed convolutional stack when the speaker embedding s and quantized log median F0 m have been zeroed out.

preserved at each intermediate layer and in the final output signal, indicating that the content information has been passed through the model properly. However, we also notice that the speaker's F0 band and its harmonics do not form at any point, indicating that the speaker's vocal characteristics have not been imparted onto the output signal. Intuitively, this makes sense; since the model does not have any conditioning information about the speaker's identity, there is no way for it to determine the characteristics of the speaker's voice.

These results demonstrate that the content-related features we feed into LVC-VC properly transfer the content information of the source utterance to the output of the model without allowing any speaker information through.

9.2.2 Content information

To analyze how LVC-VC incorporates content information, we made the model generate audio after zeroing out the low-quefrency liftered mel spectrogram $\mathbf{H}$. However, we did continue to provide the normalized F0 contour $\mathbf{p}_{\text{norm}}$ in order to see how it would interact with the speaker information s and m. Figure 9-7 illustrates the final output of the model when it generates audio with no content information, and Figure 9-8 illustrates the intermediate outputs of the transposed convolution stacks.

The outputs are essentially the reverse of what we see when we zero out the speaker

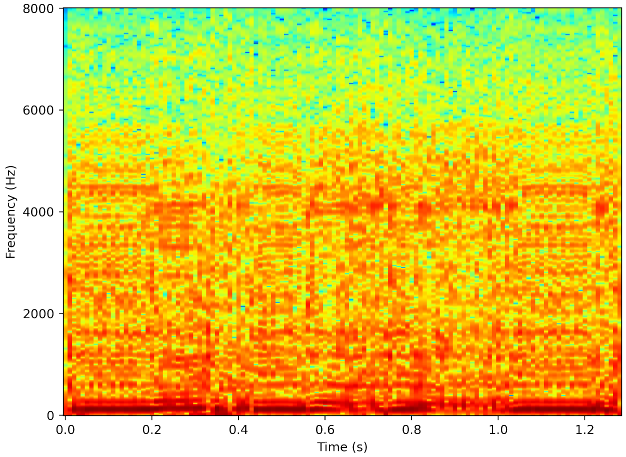
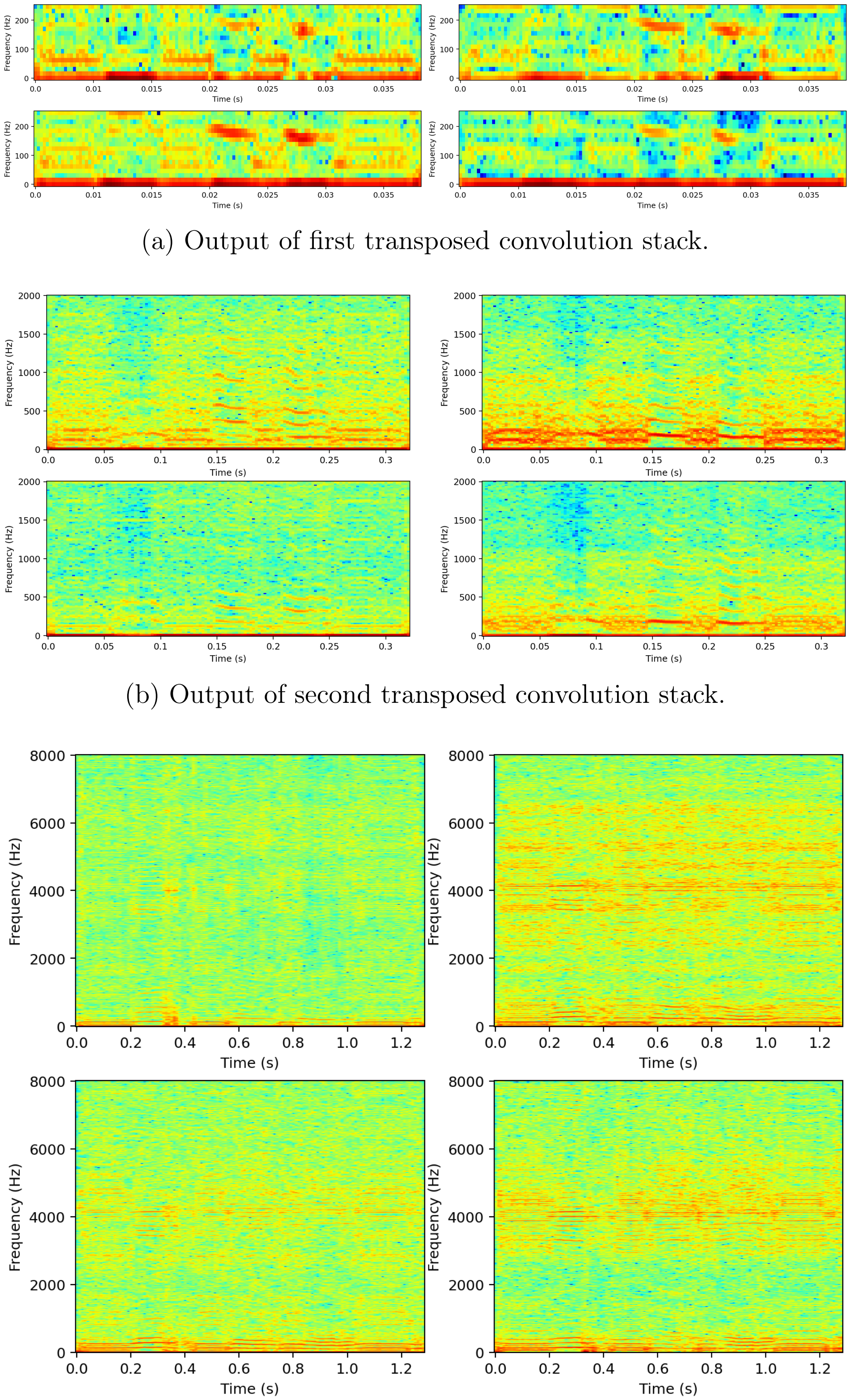


Figure 9-7: Spectrogram computed from the output signal generated by LVC-VC when the low-quefrency liftered mel spectrogram $\mathbf{H}$ has been zeroed out.



(a) Output of first transposed convolution stack.

(b) Output of second transposed convolution stack.

(c) Output of third transposed convolution stack.

Figure 9-8: Results of computing the STFT on the intermediate outputs of LVC-VC after each transposed convolutional stack when the low-quefrency liftered mel spectrogram **H** has been zeroed out.

information. Voiced segments, which are specified by the normalized F0 contour $\mathbf{p}_{\text{norm}}$, still appear to be generated more or less properly; the shape and contour of the F0 and its harmonic frequencies still form somewhat normally, demonstrating that $\mathbf{p}_{\text{norm}}$ is being successfully "un-normalized" by combining it with the speaker information. However, as expected, the spectral envelope and formants do not form at all, resulting in a situation in which the F0 contour is present but there is no content information for it to correspond to. In addition, we see that in the absence of the knowledge of which unvoiced frames are silent, the model appears to fill in the gaps with white noise-like artifacts up to around 6,000 Hz (as seen in Figure 9-7).

These results indicate that the speaker-related features that are fed into LVC-VC allow the model to effectively express the speaker's vocal characteristics in the output audio without passing any content information through.

9.3 Summary

Overall, the analyses in this chapter support the intuition that we provided earlier in Section 4.1.1: that LVC-VC combines speaker information with the content features in such a way that "un-warps" the low-quefrency liftered spectrogram $\mathbf{H}$ and "un-normalizes" the normalized F0 contour $\mathbf{p}_{\text{norm}}$. This indicates that our strategy for LVC-VC's design—combining carefully designed input features in an end-to-end vocoder-like framework, rather than training the model to explicitly disentangle and recombine the information in an utterance—is an effective way of synthesizing audio using the speaker and content information taken from different utterances.

Part IV

Wrap-Up

Chapter 10

Limitations and Future Work

In this chapter, we explore some of the main limitations of the voice conversion and anonymization methods that we have proposed, as well as extensions and future directions that could be taken to address them.

10.1 Generated Audio Quality

We saw in Chapter 8 that LVC-VC is able to generate high quality audio in terms of naturalness when the target speaker for voice conversion was seen during training, regardless of the source speaker. However, we also saw that audio quality deteriorates when the target speaker was unseen during training or arbitrarily sampled from a distribution, indicating that the way in which the model combines speaker and content information to generate audio is sensitive to the specific speaker embedding that is used. This does not preclude LVC-VC from being used for anonymization in general, as we can still produce high quality anonymized audio by converting voices to pre-specified target speakers that were seen during training. However, it does limit us from effectively using the anonymization method from Chapter 5 in practice.

We hypothesize that this issue could be mitigated by training the model on data from a much larger number of speakers. During our ablation studies, we saw that training the model on speaker embeddings sampled from GMMs of seen speakers' embeddings led to greater robustness and much better audio quality when performing zero-shot voice conversion compared to training using a single mean embedding for each speaker. This suggests that exposure to a wider space of speaker embeddings during training could help the model generalize better to arbitrary speaker embeddings during inference.

Indeed, the VCTK corpus that we used is quite small, especially for modern deep learning standards—it contains only 109 unique speakers, of which only 99 were used for training. We used this dataset primarily because of convenience; it is the standard dataset that is used in most of the voice conversion literature, and its small size makes it practical for training models in a reasonable amount of time. However, using much larger datasets such as VoxCeleb1 [65] or VoxCeleb2 [12], which have on the order of thousands of speakers, could greatly improve LVC-VC's ability to generate higher quality speech even when using previously unseen or arbitrary speaker embeddings. Larger datasets could also make it easier to train the speaker encoder E_s jointly with the rest of the model since it would then be exposed to a larger variety of speakers; this could lead to further improvements compared to using a fixed pre-trained network.

In addition, recall that audio quality degraded significantly when transforming speakers from LVN conversations, regardless of whether the target voice had been a previously seen speaker or an arbitrary sampled one. This was likely due to the significantly noisier and more reverberant nature of LVN audio, which consists of "in the wild" data that was recorded under a wide variety of uncontrolled conditions. We previously mentioned that this was not entirely surprising given that LVC-VC was never trained on anything other than clean VCTK data. Previous works on automatic speech recognition [45] and speech representation learning [88] have demonstrated the importance of data augmentation for training speech models that are robust to noise. We believe that the quality degradation issues could be somewhat mitigated by training LVC-VC on audio from more diverse sources and/or by performing data augmentation. It could also be helpful to preprocess noisy or reverberant audio with speech enhancement or denoising systems.

Finally, audio quality could also be improved by using a larger model. In our experiments, we based the generator architecture on the variant of UnivNet called

UnivNet-c16, which uses a channel size of 16 in its convolutional layers, as opposed to the larger variant UnivNet-c32, which uses a channel size of 32. Again, this was primarily for convenience reasons, as UnivNet-c16 was more lightweight and could be trained faster. However, [35] showed that UnivNet-c32 generated significantly higher quality speech than UnivNet-c16, especially when vocoding spectrograms of previously unseen speakers. Therefore, simply changing our base generator architecture to use 32 channels in its convolutional layers could improve LVC-VC's performance.

10.2 Anonymization and Voice Attribute Control

From an anonymization perspective, the capabilities of our proposed methods are limited by the current VC paradigm, which can only handle making changes to an utterance's timbre. This means that while a converted utterance might sound like it was spoken by a different person, it will still maintain certain characteristics of the original speaker that can be important personal identifiers, such as rhythm and accent. Therefore, only a limited degree of anonymization is possible through pure voice conversion, and it does not include capabilities for more fine-grained control when transforming speech.

Relatively little research has been done on methods that can flexibly and independently disentangle and alter these other characteristics of an utterance. However, some recent works have shown that it is possible to explicitly disentangle certain other factors of speech for more fine-grained control. For example, [84] demonstrated that it is possible to disentangle the rhythm, pitch, and timbre of a speech signal using specifically designed information perturbation and autoencoder information bottlenecks for more controllable speech synthesis. Meanwhile, [98] proposed a method for learning a latent space of speaker identities that allows for sampling arbitrary, non-existent voices while explicitly conditioning on locale (accent) and gender. [9] introduced a method that could alter timbre, shift pitch, and perform time-scale modification of speech by using an information perturbation strategy for various input features in a speech analysis and synthesis framework.

Going beyond the current VC paradigm, a direction of future work could be the development of a system that can change various paralinguistic attributes of a voice, and in doing so, realistically and flexibly morph a voice without the need for specifying a target speaker. One way of doing this could be by harnessing the power of deep generative models to create a speaker embedding space, rather than relying on a speaker encoder pre-trained on a recognition task. Some recent works have demonstrated the feasibility of controllable speech synthesis frameworks by sampling from the latent spaces of generative models [33, 110]. However, these methods were all used for text-to-speech applications. Extending such frameworks would have strong implications not just for the task of speaker anonymization, but for the overall field of speech synthesis.

Chapter 11

Ethical Considerations

As machine learning research and applications become more ubiquitous across society, they have raised the capabilities of many technologies and increased the likelihood of meaningful social benefit. However, they have also brought with them many uncertainties with regards to potential misuses of these technologies. Indeed, problems with data privacy, algorithmic bias, automation risk, and potential malicious uses of artificial intelligence have been well-documented [116]. These concerns are especially relevant for application-based research that aims to be deployed in the real world.

This thesis aims to improve the state-of-the-art in voice conversion in order to effectively perform the task of speaker anonymization. Unfortunately, voice conversion is a field that is fraught with potential misuse. So-called "audio deepfakes" can be used to deceive people by synthetically generating statements and attributing them to certain individuals; this can lead to harmful actions such as voice spoofing [43] or the spread of fake news and misinformation [93]. To the extent that this work enables more realistic, targeted manipulation of speaker identities, it could potentially exacerbate these misuses if used by a malicious party.

Consequently, a wide variety of recent work has sought to address the question of how to deal with audio deepfakes. These include techniques for anti-spoofing [114, 38, 101], as well as for more general fake audio detection [62, 14]. Recently, there have also been several public challenges to encourage the development of more systems that can detect fake audio, such as ASVspoof [105, 121] and the Audio Deep Synthesis Detection Challenge (ADD) [123]. Going forward, the speech machine learning community should continue to encourage these directions of research and raise awareness of the potential problems with high-fidelity synthetic audio.

There are also approaches that can be taken from a more immediately practical standpoint [69]. First, organizations that utilize synthetic voices generated by text-to-speech (TTS) or voice conversion technologies should provide adequate disclosure to audiences when they do so; this is especially important if using the voice of a well-known person. Doing this can help minimize the risk of harmful outcomes from potential deception and can also increase trust in the organization delivering the voice. When generating or converting voices to sound like real speakers, the owners of the voices should also have control over their voice model (i.e., give permission for how and where it will be used) and be compensated for their use if appropriate.

In the past, using voice conversion methods for speaker anonymization necessitated changing the original voice to a different existing person's voice. The un-targeted speaker anonymization approach introduced in Chapter 5 was our way of trying to go beyond this paradigm and avoid some of the pitfalls associated with audio deepfakes. Although there are currently some key limitations to our proposed method, we see many potential avenues to improving un-targeted voice conversion and synthesis, as described in Section 10.2. In this regard, we believe that additional research to extend and generalize the capabilities of VC technologies can help mitigate some concerns about audio deepfakes, at least in the present anonymization setting.

Chapter 12

Conclusion

12.1 Summary of Work

In this thesis, we were motivated to develop a speaker anonymization system that could effectively mask the vocal identity of spoken utterances while maintaining their prosodic elements. We chose to approach the anonymization problem from the lens of voice conversion (VC) because it would allow us to preserve the expressivity of utterances while transforming them to sound like other individuals. To this end, we presented Location-Variable Convolution-based Voice Conversion (LVC-VC), an end-to-end model for zero-shot voice conversion that is able to convert the vocal identity of an utterance to and from that of any speaker. We found that LVC-VC is able to achieve voice conversion performance that is competitive with or better than many current state-of-the-art zero-shot VC models, achieving a good balance between naturalness, intelligibility, and voice style transfer accuracy.

Furthermore, we introduced a method for extending targeted voice conversion to un-targeted voice anonymization, in which arbitrary and potentially non-existent voices are sampled from a distribution of speakers and used as the target voice in our VC model. Using regular VC methods for anonymization brings up a variety of practical and ethical issues because they necessitate that a real target speaker be specified in order to change a voice. Our proposed anonymization method aimed to avoid these concerns by eliminating the need for specifying a real target speaker.

12.2 Final Remarks

In the age of the web and social media, where text-based discourse has become so ubiquitous, spoken language still remains one of the most meaningful modes of communication that we have. Speaking directly to others and listening to voices is fundamentally powerful in a way that reading a text message, a Tweet, or a blog post is not. Thus, speech allows us to tell stories and spread ideas with others extremely effectively, perhaps to an even greater extent than we sometimes realize. Working towards a high quality method for anonymizing speech can be seen as a step towards the democratization of the voice, empowering people everywhere to speak and share their perspectives with others more freely. To this end, this work hopes to have made a small contribution in that direction.

Appendix A

Design of Amazon Mechanical Turk Surveys for Subjective Evaluations

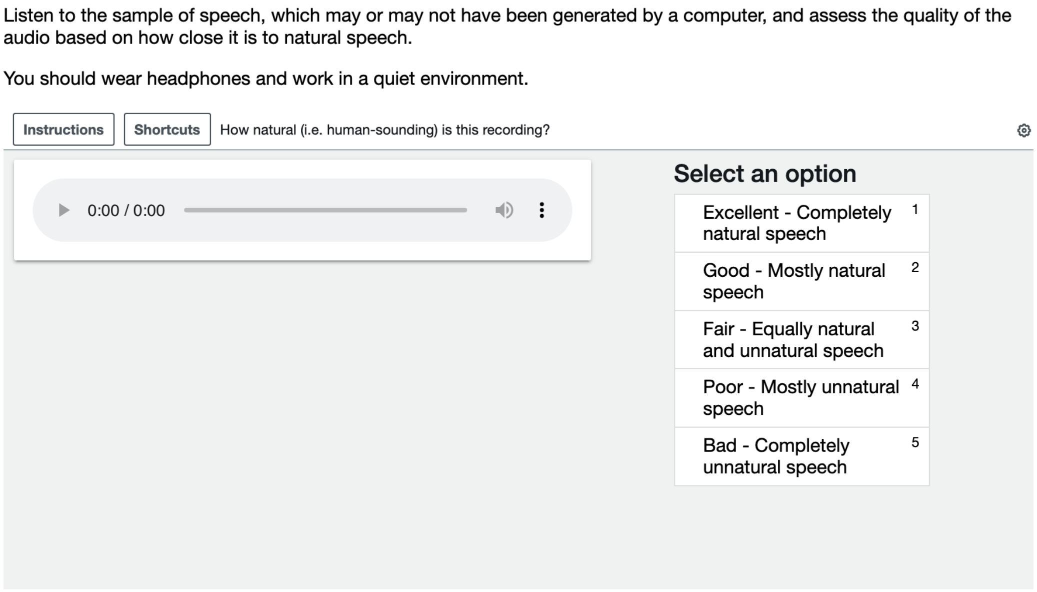
A.1 Subjective Listening Test for Naturalness (MOS)

For the subjective listening test for evaluating the naturalness of utterances, subjects were asked to assign a score from 1–5 on the naturalness of the audio. 1 meant that the utterance did not sound natural at all and 5 meant that the utterance sounded completely natural. Participants were paid $0.05 per response. The full instructions given to the subjects were as follows:

> *Listen to the sample of speech, which may or may not have been generated by a computer, and assess the quality of the audio based on how close it is to natural speech.*
>
> *You should wear headphones and work in a quiet environment.*

The rubric for evaluation was as follows:

- Excellent (5) – Completely natural speech
- Good (4) – Mostly natural speech
- Fair (3) – Equally natural and unnatural speech
- Poor (2) – Mostly unnatural speech
- Bad (1) – Completely unnatural speech


Listen to the sample of speech, which may or may not have been generated by a computer, and assess the quality of the audio based on how close it is to natural speech.

You should wear headphones and work in a quiet environment.

How natural (i.e. human-sounding) is this recording?

Select an option

Excellent - Completely natural speech	1
Good - Mostly natural speech	2
Fair - Equally natural and unnatural speech	3
Poor - Mostly unnatural speech	4
Bad - Completely unnatural speech	5

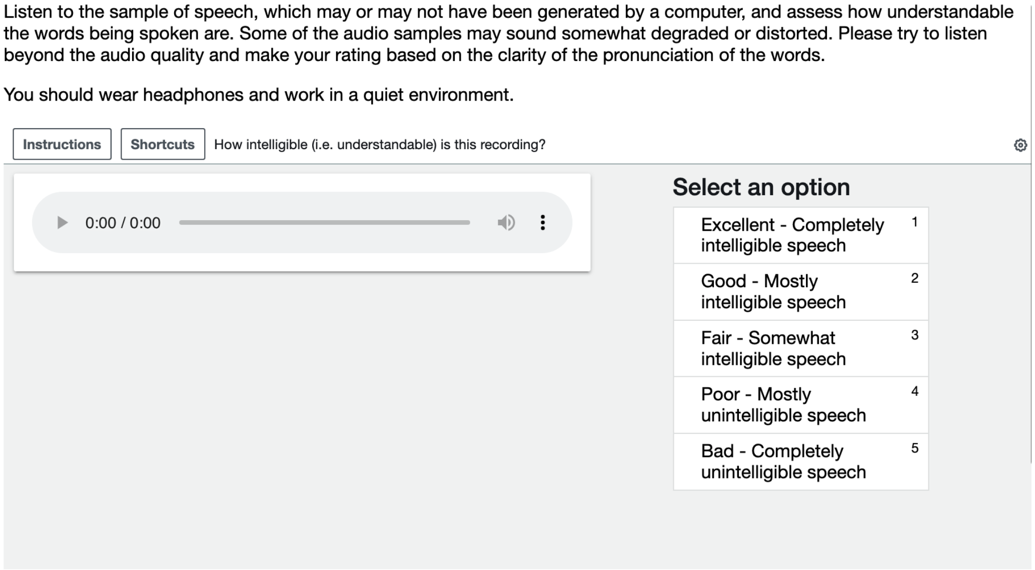
Figure A-1: Amazon Mechanical Turk instructions for subjective evaluations of naturalness.

Figure A-1 shows a screenshot of the response page for subjects.

A.2 Subjective Listening Test for Intelligibility

For the subjective listening test for evaluating the intelligibility of utterances, subjects were asked to assign a score from 1–5 on the intelligibility of the audio. 1 meant that the utterance was not understandable at all and 5 meant that the utterance was completely understandable. Participants were paid $0.05 per response. The full instructions given to the subjects were as follows:

> *Listen to the sample of speech, which may or may not have been generated by a computer, and assess how understandable the words being spoken are. Some of the audio samples may sound somewhat degraded or distorted. Please try to listen beyond the audio quality and make your rating based on the clarity of the pronunciation of the words.*
>
> *You should wear headphones and work in a quiet environment.*



Listen to the sample of speech, which may or may not have been generated by a computer, and assess how understandable the words being spoken are. Some of the audio samples may sound somewhat degraded or distorted. Please try to listen beyond the audio quality and make your rating based on the clarity of the pronunciation of the words.

You should wear headphones and work in a quiet environment.

How intelligible (i.e. understandable) is this recording?

Instructions
Shortcuts
0:00 / 0:00
Select an option
Excellent - Completely intelligible speech 1
Good - Mostly intelligible speech 2
Fair - Somewhat intelligible speech 3
Poor - Mostly unintelligible speech 4
Bad - Completely unintelligible speech 5

Figure A-2: Amazon Mechanical Turk instructions for subjective evaluations of intelligibility.

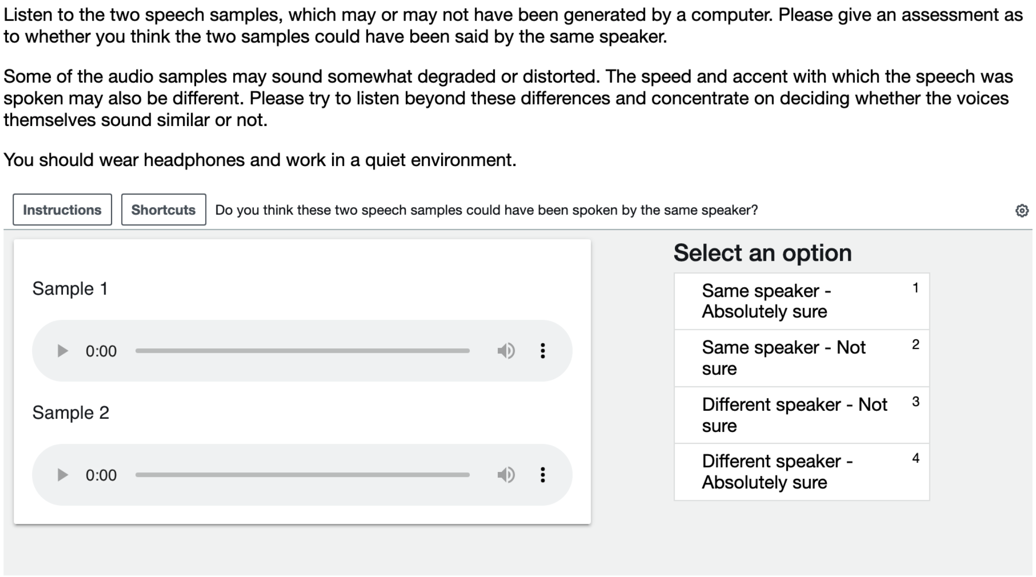
The rubric for evaluation was as follows:

- Excellent (5) – Completely intelligible speech
- Good (4) – Mostly intelligible speech
- Fair (3) – Somewhat intelligible speech
- Poor (2) – Mostly unintelligible speech
- Bad (1) – Completely unintelligible speech

Figure A-2 shows a screenshot of the response page for subjects.

A.3 Subjective Listening Test for Similarity

For the subjective listening test for evaluating the similarity of two utterances, subjects were asked to indicate whether the two voices sounded like the could have come from the same speaker. Participants were paid $0.10 per response. The full instructions given to the subjects were as follows:



Listen to the two speech samples, which may or may not have been generated by a computer. Please give an assessment as to whether you think the two samples could have been said by the same speaker.

Some of the audio samples may sound somewhat degraded or distorted. The speed and accent with which the speech was spoken may also be different. Please try to listen beyond these differences and concentrate on deciding whether the voices themselves sound similar or not.

You should wear headphones and work in a quiet environment.

Do you think these two speech samples could have been spoken by the same speaker?

Instructions
Shortcuts
Sample 1
0:00
Sample 2
0:00
Select an option
Same speaker - Absolutely sure 1
Same speaker - Not sure 2
Different speaker - Not sure 3
Different speaker - Absolutely sure 4

Figure A-3: Amazon Mechanical Turk instructions for subjective evaluations of similarity.

> *Listen to the two speech samples, which may or may not have been generated by a computer. Please give an assessment as to whether you think the two samples could have been said by the same speaker.*
>
> *Some of the audio samples may sound somewhat degraded or distorted. The speed and accent with which the speech was spoken may also be different. Please try to listen beyond these differences and concentrate on deciding whether the voices themselves sound similar or not.*
>
> *You should wear headphones and work in a quiet environment.*

The rubric for evaluation was as follows:

- Same speaker – Absolutely sure
- Same speaker – Not sure
- Different speaker – Not sure
- Different speaker – Absolutely sure

Figure A-3 shows a screenshot of the response page for subjects.

Bibliography

[1] Ranya Aloufi, Hamed Haddadi, and David Boyle. Privacy-preserving voice analysis via disentangled representations. In *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pages 1–14, 2020.

[2] Ranya Aloufi, Hamed Haddadi, and David Boyle. Configurable privacy-preserving automatic speech recognition. *arXiv preprint arXiv:2104.00766*, 2021.

[3] Bishnu S Atal and Suzanne L Hanauer. Speech analysis and synthesis by linear prediction of the speech wave. *The Journal of the Acoustical Society of America*, 50(2B):637–655, 1971.

[4] BS Atal. Determination of the vocal-tract shape directly from the speech wave. *The Journal of the Acoustical Society of America*, 47(1A):65–65, 1970.

[5] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460, 2020.

[6] Ferdinand Brasser, Tommaso Frassetto, Korbinian Riedhammer, Ahmad-Reza Sadeghi, Thomas Schneider, and Christian Weinert. Voiceguard: Secure and private speech processing. In *Interspeech*, volume 18, pages 1303–1307, 2018.

[7] 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.

[8] Yen-Hao Chen, Da-Yi Wu, Tsung-Han Wu, and Hung-yi Lee. Again-vc: A one-shot voice conversion using activation guidance and adaptive instance normalization. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5954–5958. IEEE, 2021.

[9] Hyeong-Seok Choi, Juheon Lee, Wansoo Kim, Jie Lee, Hoon Heo, and Kyogu Lee. Neural analysis and synthesis: Reconstructing speech from self-supervised representations. *Advances in Neural Information Processing Systems*, 34, 2021.

[10] Ju-chieh Chou and Hung-Yi Lee. One-shot voice conversion by separating speaker and content representations with instance normalization. *Proc. Interspeech 2019*, pages 664–668, 2019.

[11] 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.

[12] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018.

[13] Alice Cohen-Hadria, Mark Cartwright, Brian McFee, and Juan Pablo Bello. Voice anonymization in urban sound recordings. In *2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2019.

[14] 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.

[15] Daniel Erro, Asunción Moreno, and Antonio Bonafonte. Inca algorithm for training voice conversion systems from nonparallel corpora. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(5):944–953, 2009.

[16] Fuming Fang, Xin Wang, Junichi Yamagishi, Isao Echizen, Massimiliano Todisco, Nicholas Evans, and Jean-Francois Bonastre. Speaker anonymization using x-vector and neural waveform models. *arXiv preprint arXiv:1905.13561*, 2019.

[17] Gunnar Fant. *Acoustic theory of speech production.* Number 2. Walter de Gruyter, 1970.

[18] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients - how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems*, volume 33, pages 16937–16947, 2020.

[19] 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. *Advances in Neural Information Processing Systems*, 30, 2017.

[20] Tobias Glasmachers. Limits of end-to-end learning. In *Asian Conference on Machine Learning*, pages 17–32. PMLR, 2017.

[21] Félix Gontier, Mathieu Lagrange, Catherine Lavandier, and Jean-François Petiot. Privacy aware acoustic scene synthesis using deep spectral feature inversion. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 886–890. IEEE, 2020.

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.

[23] 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.

[24] Yaowei Han, Sheng Li, Yang Cao, Qiang Ma, and Masatoshi Yoshikawa. Voice-indistinguishability: Protecting voiceprint in privacy-preserving speech data release. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.

[25] Kei Hashimoto, Junichi Yamagishi, and Isao Echizen. Privacy-preserving sound to degrade automatic speaker verification performance. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5500–5504. IEEE, 2016.

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[27] Elina Helander, Jan Schwarz, Jani Nurminen, Hanna Silen, and Moncef Gabbouj. On the impact of alignment on voice conversion performance. In *Ninth Annual Conference of the International Speech Communication Association*, 2008.

[28] Elina Helander, Tuomas Virtanen, Jani Nurminen, and Moncef Gabbouj. Voice conversion using partial least squares regression. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(5):912–921, 2010.

[29] Hee Soo Heo, Bong-Jin Lee, Jaesung Huh, and Joon Son Chung. Clova baseline system for the voxceleb speaker recognition challenge 2020. *arXiv preprint arXiv:2009.14153*, 2020.

[30] Daniel Hirst and Albert Di Cristo. A survey of intonation systems. *Intonation Systems: A Survey of Twenty Languages*, 144, 1998.

[31] Eric Horvitz and Deirdre Mulligan. Data, privacy, and the greater good. *Science*, 349(6245):253–255, 2015.

[32] Chin-Cheng Hsu, Hsin-Te Hwang, Yi-Chiao Wu, Yu Tsao, and Hsin-Min Wang. Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks. *arXiv preprint arXiv:1704.00849*, 2017.

[33] Wei-Ning Hsu, Yu Zhang, Ron J Weiss, Heiga Zen, Yonghui Wu, Yuxuan Wang, Yuan Cao, Ye Jia, Zhifeng Chen, Jonathan Shen, et al. Hierarchical generative modeling for controllable speech synthesis. In *International Conference on Learning Representations*, 2018.

[34] Jim Isaak and Mina J Hanna. User data privacy: Facebook, cambridge analytica, and privacy protection. *Computer*, 51(8):56–59, 2018.

[35] Won Jang, Dan Lim, Jaesam Yoon, Bongwan Kim, and Juntae Kim. UnivNet: A Neural Vocoder with Multi-Resolution Spectrogram Discriminators for High-Fidelity Waveform Generation. In *Proc. Interspeech 2021*, pages 2207–2211, 2021.

[36] Qin Jin, Arthur R Toth, Tanja Schultz, and Alan W Black. Speaker deidentification via voice transformation. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 529–533. IEEE, 2009.

[37] 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.

[38] 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.

[39] Hirokazu Kameoka, Takuhiro Kaneko, Kou Tanaka, and Nobukatsu Hojo. Stargan-vc: Non-parallel many-to-many voice conversion using star generative adversarial networks. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 266–273. IEEE, 2018.

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

[41] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. Cyclegan-vc2: Improved cyclegan-based non-parallel voice conversion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6820–6824. IEEE, 2019.

[42] Takuhiro Kaneko, Hirokazu Kameoka, Kou Tanaka, and Nobukatsu Hojo. Stargan-vc2: Rethinking conditional methods for stargan-based voice conversion. In *INTERSPEECH*, 2019.

[43] Tomi Kinnunen, Zhi-Zheng Wu, Kong Aik Lee, Filip Sedlak, Eng Siong Chng, and Haizhou Li. Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4401–4404. IEEE, 2012.

[44] Media Knight Commission on Trust and Democracy. *Crisis in Democracy: Renewing Trust in America: the Report of the Knight Commission on Trust, Media and Democracy.* Aspen Institute, 2019.

[45] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. Audio augmentation for speech recognition. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[46] Kazuhiro Kobayashi, Wen-Chin Huang, Yi-Chiao Wu, Patrick Lumban Tobing, Tomoki Hayashi, and Tomoki Toda. crank: An open-source software for non-parallel voice conversion based on vector-quantized variational autoencoder. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5934–5938. IEEE, 2021.

[47] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33, 2020.

[48] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[49] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in Neural Information Processing Systems*, 32, 2019.

[50] Sang-Hoon Lee, Ji-Hoon Kim, Hyunseung Chung, and Seong-Whan Lee. Voicemixer: Adversarial voice style mixup. *Advances in Neural Information Processing Systems*, 34, 2021.

[51] David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau. Federated learning for keyword spotting. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6341–6345. IEEE, 2019.

[52] Li-Juan Liu, Zhen-Hua Ling, Yuan Jiang, Ming Zhou, and Li-Rong Dai. Wavenet vocoder with limited training data for voice conversion. In *Interspeech*, pages 1983–1987, 2018.

[53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

[54] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30, page 3. Citeseer, 2013.

[55] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

[56] Matthias Mauch and Simon Dixon. pyin: A fundamental frequency estimator using probabilistic threshold distributions. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 659–663. IEEE, 2014.

[57] Phil McAleer, Alexander Todorov, and Pascal Belin. How do you say 'hello'? personality impressions from brief novel voices. *PloS One*, 9(3):e90779, 2014.

[58] Siobhán McHugh. The affective power of sound: oral history on radio. *The Oral History Review*, 2019.

[59] MIT Center for Constructive Communication. Madison Police and Fire Commission Case Study. https://cortico.ai/wp-content/uploads/2021/08/madison-case-study-.pdf, 2020.

[60] MIT Center for Constructive Communication. RealTalk for Change Boston. https://www.media.mit.edu/projects/realtalk-for-change-boston/overview/, 2021.

[61] Gabriel Mittag, Babak Naderi, Assmaa Chehadi, and Sebastian Möller. Nisqa: A deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. *Proc. Interspeech 2021*, pages 2127–2131, 2021.

[62] Trisha Mittal, Uttaran Bhattacharya, Rohan Chandra, Aniket Bera, and Dinesh Manocha. Emotions don't lie: An audio-visual deepfake detection method using affective cues. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2823–2832, 2020.

[63] 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.

[64] Philip McCord Morse and K Uno Ingard. *Theoretical acoustics*. Princeton University Press, 1986.

[65] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612*, 2017.

[66] Andreas Nautsch, Catherine Jasserand, Els Kindt, Massimiliano Todisco, Isabel Trancoso, and Nicholas Evans. The gdpr and speech data: Reflections of the legal and technology communities: First steps towards a common understanding. In *Interspeech: Crossroads of Speech and Language*, 2019.

[67] Andreas Nautsch, Abelino Jiménez, Amos Treiber, Jascha Kolberg, Catherine Jasserand, Els Kindt, Héctor Delgado, Massimiliano Todisco, Mohamed Amine Hmani, Aymen Mtibaa, et al. Preserving privacy in speaker and speech characterisation. *Computer Speech & Language*, 58:441–480, 2019.

[68] Bac Nguyen and Fabien Cardinaux. Nvc-net: End-to-end adversarial voice conversion. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7012–7016. IEEE, 2022.

[69] Ben Noa Moussa and Eric Urban. Guidelines for responsible deployment of synthetic voice technology. https://docs.microsoft.com/en-us/legal/cognitive-services/speech-service/custom-neural-voice/concepts-guidelines-responsible-deployment-synthetic, 2022.

[70] Koji Okabe, Takafumi Koshinaka, and Koichi Shinoda. Attentive statistics pooling for deep speaker embedding. *Proc. Interspeech 2018*, pages 2252–2256, 2018.

[71] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[72] Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. In *International Conference on Machine Learning*, pages 3918–3926. PMLR, 2018.

[73] 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.

[74] Sarah Oppold and Melanie Herschel. A system framework for personalized and transparent data-driven decisions. In *International Conference on Advanced Information Systems Engineering*, pages 153–168. Springer, 2020.

[75] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[76] Manas A Pathak, Bhiksha Raj, Shantanu D Rane, and Paris Smaragdis. Privacy-preserving speech processing: cryptographic and string-matching frameworks show promise. *IEEE Signal Processing Magazine*, 30(2):62–74, 2013.

[77] Jose Patino, Natalia Tomashenko, Massimiliano Todisco, Andreas Nautsch, and Nicholas Evans. Speaker anonymisation using the mcadams coefficient. *arXiv preprint arXiv:2011.01130*, 2020.

[78] Wei Ping, Kainan Peng, and Jitong Chen. Clarinet: Parallel wave generation in end-to-end text-to-speech. In *International Conference on Learning Representations*, 2018.

[79] Wei Ping, Kainan Peng, Kexin Zhao, and Zhao Song. Waveflow: A compact flow-based model for raw audio. In *International Conference on Machine Learning*, pages 7706–7716. PMLR, 2020.

[80] Miran Pobar and Ivo Ipšić. Online speaker de-identification using voice transformation. In *2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 1264–1267. IEEE, 2014.

[81] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[82] Jianwei Qian, Haohua Du, Jiahui Hou, Linlin Chen, Taeho Jung, Xiang-Yang Li, Yu Wang, and Yanbo Deng. Voicemask: Anonymize and sanitize voice input on mobile devices. *arXiv preprint arXiv:1711.11460*, 2017.

[83] Kaizhi Qian, Zeyu Jin, Mark Hasegawa-Johnson, and Gautham J Mysore. F0-consistent many-to-many non-parallel voice conversion via conditional autoencoder. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6284–6288. IEEE, 2020.

[84] Kaizhi Qian, Yang Zhang, Shiyu Chang, Mark Hasegawa-Johnson, and David Cox. Unsupervised speech decomposition via triple information bottleneck. In *International Conference on Machine Learning*, pages 7836–7846. PMLR, 2020.

[85] Kaizhi Qian, Yang Zhang, Shiyu Chang, Jinjun Xiong, Chuang Gan, David Cox, and Mark Hasegawa-Johnson. Global prosody style transfer without text transcriptions. In *International Conference on Machine Learning*, pages 8650–8660. PMLR, 2021.

[86] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *International Conference on Machine Learning*, pages 5210–5219. PMLR, 2019.

[87] Aloufi Ranya, Haddadi Hamed, and Boyle David. Emotionless: Privacy-preserving speech analysis for voice assistants. In *Privacy Preserving in Machine Learning (CCS19) Workshop, London, UK*, 2019.

[88] 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 *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6989–6993. IEEE, 2020.

[89] Klaus R Scherer, Rainer Banse, and Harald G Wallbott. Emotion inferences from vocal expression correlate across languages and cultures. *Journal of Cross-Cultural Psychology*, 32(1):76–92, 2001.

[90] Juliana Schroeder and Nicholas Epley. The sound of intellect: Speech reveals a thoughtful mind, increasing a job candidate's appeal. *Psychological Science*, 26(6):877–891, 2015.

[91] Joan Serrà, Santiago Pascual, and Carlos Segura Perales. Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. *Advances in Neural Information Processing Systems*, 32:6793–6803, 2019.

[92] 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.

[93] Shivangi Singhal, Rajiv Ratn Shah, Tanmoy Chakraborty, Ponnurangam Kumaraguru, and Shin'ichi Satoh. Spotfake: A multi-modal framework for fake news detection. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pages 39–47. IEEE, 2019.

[94] Berrak Sisman, Junichi Yamagishi, Simon King, and Haizhou Li. An overview of voice conversion and its challenges: From statistical modeling to deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[95] Brij Mohan Lal Srivastava, Aurélien Bellet, Marc Tommasi, and Emmanuel Vincent. Privacy-preserving adversarial representation learning in asr: Reality or illusion? In *INTERSPEECH*, 2019.

[96] Brij Mohan Lal Srivastava, Natalia Tomashenko, Xin Wang, Emmanuel Vincent, Junichi Yamagishi, Mohamed Maouche, Aurélien Bellet, and Marc Tommasi. Design choices for x-vector based speaker anonymization. In *INTERSPEECH 2020*, 2020.

[97] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[98] Daisy Stanton, Matt Shannon, Soroosh Mariooryad, RJ Skerry-Ryan, Eric Battenberg, Tom Bagby, and David Kao. Speaker generation. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7897–7901. IEEE, 2022.

[99] Lifa Sun, Shiyin Kang, Kun Li, and Helen Meng. Voice conversion using deep bidirectional long short-term memory based recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4869–4873. IEEE, 2015.

[100] David Sundermann, Harald Hoge, Antonio Bonafonte, Hermann Ney, Alan Black, and Shri Narayanan. Text-independent voice conversion based on unit selection. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 1, pages I–I. IEEE, 2006.

[101] Hemlata Tak, Jose Patino, Massimiliano Todisco, Andreas Nautsch, Nicholas Evans, and Anthony Larcher. End-to-end anti-spoofing with rawnet2. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6369–6373. IEEE, 2021.

[102] Ryoichi Takashima, Tetsuya Takiguchi, and Yasuo Ariki. Exemplar-based voice conversion in noisy environment. In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 313–317. IEEE, 2012.

[103] D. Tannen, C.N. Li, S.A. Thompson, P.L.D. Tannen, R.O. Freedle, S.B. Heath, G.M. Green, J. Goody, A. Hildyard, D.R. Olson, et al. *Spoken and Written Language: Exploring Orality and Literacy*. Advances in Discourse Processes. ABLEX Publishing Corporation, 1982.

[104] Tomoki Toda, Alan W Black, and Keiichi Tokuda. Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2222–2235, 2007.

[105] Massimiliano Todisco, Xin Wang, Ville Vestman, Md Sahidullah, Héctor Delgado, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Tomi Kinnunen, and Kong Aik Lee. Asvspoof 2019: Future horizons in spoofed and fake audio detection. In *INTERSPEECH 2019-20th Annual Conference of the International Speech Communication Association*, 2019.

[106] Natalia Tomashenko, Brij Mohan Lal Srivastava, Xin Wang, Emmanuel Vincent, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Jose Patino, J-F Bonastre, Paul-Gauthier Noé, et al. Introducing the voiceprivacy initiative. In *INTERSPEECH 2020*, 2020.

[107] Natalia Tomashenko, Brij Mohan Lal Srivastava, Xin Wang, Emmanuel Vincent, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Jose Patino, Jean-François Bonastre, Paul-Gauthier Noé, et al. The voiceprivacy 2020 challenge evaluation plan, 2020.

[108] Natalia Tomashenko, Xin Wang, Xiaoxiao Miao, Hubert Nourtel, Pierre Champion, Massimiliano Todisco, Emmanuel Vincent, Nicholas Evans, Junichi Yamagishi, and Jean François Bonastre. The voiceprivacy 2022 challenge evaluation plan. *arXiv preprint arXiv:2203.12468*, 2022.

[109] Natalia Tomashenko, Xin Wang, Emmanuel Vincent, Jose Patino, Brij Mohan Lal Srivastava, Paul-Gauthier Noé, Andreas Nautsch, Nicholas Evans, Junichi Yamagishi, Benjamin O'Brien, et al. The voiceprivacy 2020 challenge: Results and findings. *arXiv preprint arXiv:2109.00648*, 2021.

[110] Rafael Valle, Kevin J Shih, Ryan Prenger, and Bryan Catanzaro. Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis. In *International Conference on Learning Representations*, 2020.

[111] Nal Van den Oord, Aaron anModed Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in Neural Information Processing Systems*, 29, 2016.

[112] Benjamin van Niekerk, Leanne Nortje, and Herman Kamper. Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge. In *INTERSPEECH*, 2020.

[113] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10:3152676, 2017.

[114] Qian Wang, Xiu Lin, Man Zhou, Yanjiao Chen, Cong Wang, Qi Li, and Xiangyang Luo. Voicepop: A pop noise based anti-spoofing system for voice authentication on smartphones. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2062–2070. IEEE, 2019.

[115] Mirjam Wester, Zhizheng Wu, and Junichi Yamagishi. Analysis of the voice conversion challenge 2016 evaluation results. *Interspeech 2016*, pages 1637–1641, 2016.

[116] Jess Whittlestone, Rune Nyrup, Anna Alexandrova, Kanta Dihal, and Stephen Cave. Ethical and societal implications of algorithms, data, and artificial intelligence: a roadmap for research. *London: Nuffield Foundation*, 2019.

[117] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.

[118] Zhizheng Wu, Tuomas Virtanen, Eng Siong Chng, and Haizhou Li. Exemplar-based sparse representation with residual compensation for voice conversion. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1506–1521, 2014.

[119] Feng-Long Xie, Frank K Soong, and Haifeng Li. A kl divergence and dnn-based approach to voice conversion without parallel training sentences. In *Interspeech*, pages 287–291, 2016.

[120] Junichi Yamagishi, Christophe Veaux, Kirsten MacDonald, et al. Cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit (version 0.92). 2019.

[121] Junichi Yamagishi, Xin Wang, Massimiliano Todisco, Md Sahidullah, Jose Patino, Andreas Nautsch, Xuechen Liu, Kong Aik Lee, Tomi Kinnunen, Nicholas Evans, et al. Asvspoof 2021: accelerating progress in spoofed and deepfake speech detection. In *ASVspoof 2021 Workshop-Automatic Speaker Verification and Spoofing Coutermeasures Challenge*, 2021.

[122] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE, 2020.

[123] 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. *arXiv preprint arXiv:2202.08433*, 2022.

[124] Siyang Yuan, Pengyu Cheng, Ruiyi Zhang, Weituo Hao, Zhe Gan, and Lawrence Carin. Improving zero-shot voice style transfer via disentangled representation learning. In *International Conference on Learning Representations*, 2020.

[125] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. *Proc. Interspeech 2019*, pages 1526–1530, 2019.

[126] Zhen Zeng, Jianzong Wang, Ning Cheng, and Jing Xiao. Lvcnet: Efficient condition-dependent modeling network for waveform generation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6054–6058. IEEE, 2021.

[127] Jing-Xuan Zhang, Li-Juan Liu, Yan-Nian Chen, Ya-Jun Hu, Yuan Jiang, Zhen-Hua Ling, and Li-Rong Dai. Voice conversion by cascading automatic speech recognition and text-to-speech synthesis with prosody transfer. In *Proc. Joint Workshop for the Blizzard Challenge and Voice Conversion Challenge 2020*, pages 121–125, 2020.

[128] Shi-Xiong Zhang, Yifan Gong, and Dong Yu. Encrypted speech recognition using deep polynomial networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5691–5695. IEEE, 2019.