Deep learning text-to-speech synthesis with Flowtron and WaveGlow

Veera Sairanen

School of Science

Thesis submitted for examination in partial fulfilment of the requirements for the degree of Master of Science in Technology.

Graz, 24.04.2023

Supervisor

Prof. Matias Palva, PhD

Advisor

Prof. Gernot Müller-Putz, PhD

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Veera Sairanen

Title Deep learning text-to-speech synthesis with Flowtron and WaveGlow

Degree programme Life Science Technologies

Major Biomedical Engineering **Code of major** SCI3059

Supervisor Prof. Matias Palva, PhD

Advisor Prof. Gernot Müller-Putz, PhD

Date 24.04.2023 **Number of pages** 48+7 **Language** English

Abstract

Innovation in the field of artificial speech synthesis using deep learning has been rapidly increasing over the past years. Current interest lies in the synthesis of speech that is able to model the complex prosody and stylistic features of natural spoken language using a minimal amount of data. Not only are such models remarkable from a technological perspective, they also have immense potential as an application of custom voice assistive technology (AT) for people living with speech impairments. However, more research should be focused on the evaluation of the applicability of deep learning text-to-speech (TTS) systems in a real-world context. This thesis aims to further this research by employing two well-known TTS frameworks, Flowtron and WaveGlow, to train a voice clone model on limited personal speech data of a person living with locked in syndrome (LIS). The resulting artificial voice is assessed based on human perception. In addition, the results of the model are showcased in a user-friendly TTS application that also acts as a prototype for custom voice AT. Through the work in this thesis we explore the fascinating world of deep learning based artificial speech synthesis and inspire further research in its relevance toward the development of inclusive technology.

Keywords TTS, voice cloning, deep learning, assistive technology,

Contents

Abbreviations

NN	neural network
DNN	deep neural network
TTS	text-to-speech
AT	assistive technology
LIS	locked in syndrome
BCI	brain-computer interface
MOS	mean-opinion score
EEG	electroencephalography
ERP	event-related potential
fMRI	functional magnetic resonance imaging
fNIRS	functional near-infrared spectroscopy
F0	fundamental frequency
STFT	short-time Fourier transform
HMM	hidden Markov model
SPSS	statistical parametric speech synthesis
GMM	Gaussian mixture model
MAF	masked autoregressive flow
RNN	recurrent neural network
LSTM	long short-term memory
GUI	graphical user interface
cuDNN	CUDA deep neural network
UMAP	inform manifold approximation and density
FFT	fast Fourier transform
NLL	negative log likelihood

List of Figures

List of Tables

1 Introduction

Background. In recent years we have seen a strong increase in the use of deep learning models in multiple technological applications. In the case of text-to-speech (TTS), the synthesis of artificial speech from text, deep learning has had an especially profound impact. In comparison to traditional concatenative speech synthesis methods, voice clones trained using deep learning methods offer a much more natural-sounding, human-like result and have since become predominant in the field. Deep neural networks (DNNs) are trained using recorded speech segments and their associated transcriptions and as such require minimal manual annotation and domain expertise. A typical deep learning TTS model, such as Tacotron [1], requires tens of hours of professionally recorded speech with minimal noise, correct pronunciation, a rich vocabulary, and expressive intonation. Such a large amount of data is often not feasible for most applications. Therefore current research has been focused on models which can incorporate new speakers using only a small amount of data. A common approach is to fine-tune the speaker embedding layer of a multispeaker TTS model already pretrained on a larger dataset in order to adapt the model to the voice characteristics of the target speaker [2]. Voice cloning can also operate on a zero-shot setting, in which the model can directly incorporate voices unseen during training with only a few seconds of reference speech [3], though this method tends to produce unnatural prosody and introduce signal artefacts [4]. The concept of modeling stylistic features of speech that capture emotion or prosody has also garnered much interest in the TTS field. Most research is centered around learning an embedding of speech style [5] or the representation of stylistic features in latent space [6]. However, due to the complexity of human speech signals, it is unclear to what extent these methods can accurately portray prosody and style in the way that humans perceive speech.

Significance. There are a large range of applications for high-quality artificial speech, especially as an aid for people with disabilities. People with speech impairments can communicate with others in a more natural way with the help of deep learning TTS systems. The main motivation behind this thesis is the development of customized voice assistive technology (AT) with a focus on people with locked in syndrome (LIS). People with LIS cannot move or speak due to paralysis of voluntary muscles, with the exception of certain eye muscles. People with LIS are both aware and conscious, which calls for alternative channels of communication. Existing technologies such as [7] and [8] can translate eye movements into words or commands. Brain Computer Interface (BCI) systems have also been shown to accurately decode a person's intention using brain signals [9]. Through the application of a high-quality custom voice clone, combined with eye-tracking or BCI technology, people with LIS, or other speech impairments, may regain the ability to communicate with others in a natural way with their own unique voice.

Knowledge gap. Custom voice AT would ideally sound very similar to the target speaker and as natural as possible with the possibility to express different emotions.

Flowtron [6] and WaveGlow [10] are well-known flow-based deep learning networks which perform text to mel-spectrogram and mel-spectrogram to audio synthesis, respectively. Flowtron offers control over expressiveness and style through manipulation of the latent space representation of speech. WaveGlow is considered a universal vocoder, meaning that it can produce speech from mel-spectrograms which it has not been explicitly trained on [10]. The voice clone samples introduced in the paper [6] are generated by Flowtron models trained on recordings by professional voice actors in a studio environment and synthesized into audio by a pretrained single speaker WaveGlow model. It is unclear whether the framework would provide satisfactory results when trained on personal speech recordings, which are, in most cases, lacking in both duration and quality in comparison. Since custom voice AT is centered around communication between humans, it would be important to conduct a comprehensive evaluation of how the characteristics of the cloned voice are perceived by listeners. The Flowtron and WaveGlow framework has potential in the field of AT, but more research should be conducted to ensure its compatibility to both limited reference speech and human perception.

Problem Statement. As an up and coming field, deep learning TTS systems often focus on the novelty of new methods while overlooking their real-world applicability. In order to understand the prospects of TTS, it is necessary to bridge the gap between knowledge and practice.

Aim. The primary goal of this thesis is to employ the Flowtron and WaveGlow framework to create a voice clone trained on past speech recordings of a person living with LIS. We then subsequently analyze and evaluate the potential of the framework for use as an alternative communication channel for people living with speech impairments.

Hypothesis. It is likely that the training data available for this project is not sufficient for training a Flowtron model from scratch, but would produce satisfactory results with the fine-tuning method. Further, the quality of the target speaker data may influence the quality of the resulting voice clone.

Approach. We first optimize our training setup to our speech data by visually monitoring training and validation loss progression over time. After training the final voice clone model, we generate samples with varying levels of expressiveness and speech styles and analyze the results qualitatively. We then conduct a Mean Opinion Score (MOS) listening test in order to attain a subjective assessment of the voice clone from various aspects in a real-world context. In addition, we create a simple speech synthesis computer application which generates cloned audio from text input. The application showcases the results of this thesis while also serving as an early-stage prototype of an AT communication device.

Objectives. Through the work conducted in this thesis we hope to generate awareness and inspire further research related to the important topic of inclusive technology in the context of people living with speech impairments. We also aim to gain a deeper understanding of current state-of-the-art voice cloning methods.

2 Background

2.1 Phonetics and theory of human speech production

Phonetics is a field of linguistics which studies the sounds of spoken language. Phonetics is divided into three branches: articulatory, acoustic, and auditory phonetics. They aim to describe the production, transmission, and perception of human speech sounds, respectively. Before one may embark on the complex task of modeling human speech, one must have a general understanding of the physical components of human sound generation, and how resulting sounds are interpreted by other humans as language.

2.1.1 Articulatory phonetics

Articulatory phonetics describes how the vocal tract and articulatory system components generate sounds. From a physical point of view, speech occurs as a result of muscles contracting, which push air from the lungs to the mouth and nose, and modify the air flow in order to create different sounds. The main energy source are the lungs and diaphragm. These muscles force air through a V-shaped opening between the vocal cords and the larynx, called the glottis. After the larynx, the air enters the vocal tract and exits through the mouth or nose into the atmosphere. [11]

While air travels upward, the shape of the vocal tract is altered by the configuration of and the interaction between articulatory system components, called articulators, so as to create different sounds. A diagram of the vocal tract and articulators is shown in Figure 1. Speech sounds can roughly be classified into consonants and vowels, as well as voiced and unvoiced sounds. A vowel is a sound formed without constriction to the vocal tract, while consonants are formed through partial or complete constriction. In order to create voiced sounds, the voice cords vibrate rapidly, which produces a buzzing sound. The airflow is then modified via the articulators to produce different vowel sounds. Unvoiced sounds are formed without vibration of the vocal cords. All vowels are considered voiced, while consonants can be either. [11]

Since consonant sounds are characterized by some sort of constriction to the airflow, they are also defined by place and manner of said constriction. Active articulators are the components which move to form the articulation, while passive articulators stay stationary. Examples of active articulators include the lips and tongue, and passive articulators include the roof of the mouth and teeth. Labial consonants are formed with the lips, either by pressing both lips together, or via contact between the lips and the teeth. Coronal consonants are formed with the tip of the tongue, and depending on the passive articulator location and posture of the tongue, can create a wide variety of different sounds. Dorsal consonants use the whole surface of the tongue, while radical consonants use the root of the tongue. Glottal consonants are formed via the vocal folds in the larynx. The way in which airflow is constricted,

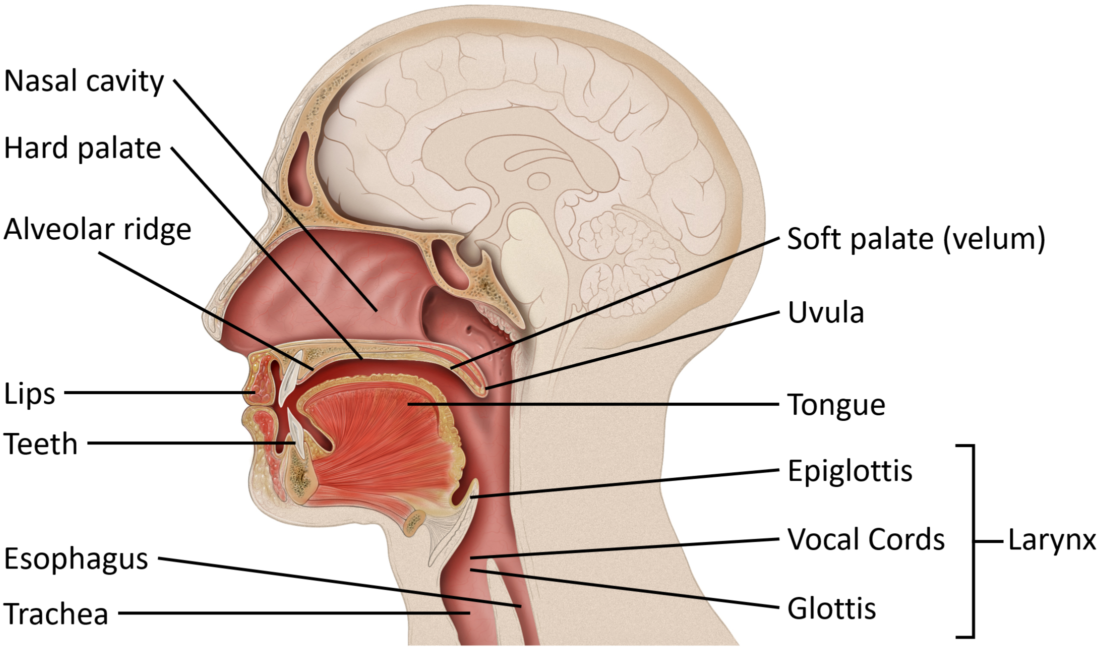


Figure 1: Diagram of vocal tract components (articulators). Image from [12]

partially or fully, affects the sound created. Plosives are formed when the airstream is fully constricted and suddenly released, resulting in a short burst of sound, while fricatives only partially obstruct the stream, creating turbulence. Trills involve the vibration of the tongue or lips. A vast assortment of vocal tract modifications via the articulators enable the generation of a broad assortment of sounds. [13]

2.1.2 Acoustic phonetics

Acoustic phonetics studies the acoustic properties of human speech sounds. Modeling continuous speech is not an easy task because of the interactions between complex audio signals. The four most important properties to investigate are frequency, time, amplitude, and formant. There are two types of frequencies in human sound: fundamental frequency (F0) and formant frequencies. F0 is formed by the oscillations of the vocal cords, the rate of which is dependent on their length, thickness, and tension. Therefore, it differs for people of different ages and genders and is closely related to pitch, which is defined as the human perception of F0. [14] Formant frequency, on the other hand, relates to the configuration of the articulators in the vocal tract. [15] Formants are acoustic energy peaks around a certain frequency in the spectrum of a sound. Each formant is associated with a particular resonance in the vocal tract. They are labeled numerically in succession from the lowest frequency to the highest after the fundamental frequency (F0). Each distinct unit of sound has corresponding formants, and can as such be considered as filters, which attenuate certain frequency components and strengthen others. [16] Sounds can thus be identified more easily by their formants and their transitions instead of by their waveforms. Formant patterns can be easily identified through spectrograms, which represent how frequencies of a signal change over time. It is obtained by splitting the audio into overlapping windows and performing the short time Fourier transform (STFT) on each window. Figure 2 shows the spectrogram of a man speaking three different syllables: “dee”, “daa”, and “doo”. As can be seen in the figure, each vowel has its own distinct spectral pattern.

2.1.3 Auditory phonetics

Auditory phonetics is concerned with how speech sounds are perceived, including how the auditory system responds to speech stimuli. The manner in which we communicate using speech sounds is very unique to humans. While many other animal species also communicate with different sounds, humans are able to convey a much wider spectrum of complex topics and emotions using speech. [18] Sounds travel into the ear as air pressure waves until they reach the eardrum. Movement of the eardrum from the waves causes vibration of three small bones in the middle ear. This vibration is amplified and transmitted to the cochlea. Inside the cochlea is the organ of corti resting atop the basilar membrane. Vibrations at different frequencies displace the membrane at different frequency-sensitive locations [19]. This in turn causes specific hair cells in the organ of corti to convert the vibration into an electrical signal. The signals are transferred to the central auditory nuclei through the auditory nerve. [20] Sound information is decoded at each stage of the auditory pathway and projected in the auditory cortex. The representation of sounds in the auditory system is tonotopic, meaning that the signal is decomposed into frequency components and analyzed separately. Different frequencies are represented in different populations of neurons. While this tonotopic organization exists throughout the entire auditory system, the mapping becomes more complicated and includes more stimuli categories in the auditory cortex. Another feature of sound perception in the auditory system is nonlinear suppression. When there is a strong activation in one group of neurons representing a certain frequency or acoustic feature, it suppresses activity in another group of neurons. This feature is important when perceiving the spectral content of sounds with a range of sound levels. [19] From an acoustic perspective, speech is simply a stream of sound with different patterns of frequency and amplitude. Auditorily, though, speech (in a familiar language) sounds segmented. One can discern words, which give meaning to the speech. One can also perceive the sound units which make up words. [21] Phones are defined as any distinct type of speech sound that exists in a language, without regard to its significance to the meaning of words. Phonemes, on the other hand, are sound units which have semantic significance and directly affect the meanings of words. Allophones are variations of phonemes which do not affect the meaning when swapped amongst eachother within a word. The table in Figure 3 presents all of the phonemes found in the English language, divided into vowels and consonants, as well as unvoiced and voiced phonemes. The relationship between phoneme perception when hearing speech and

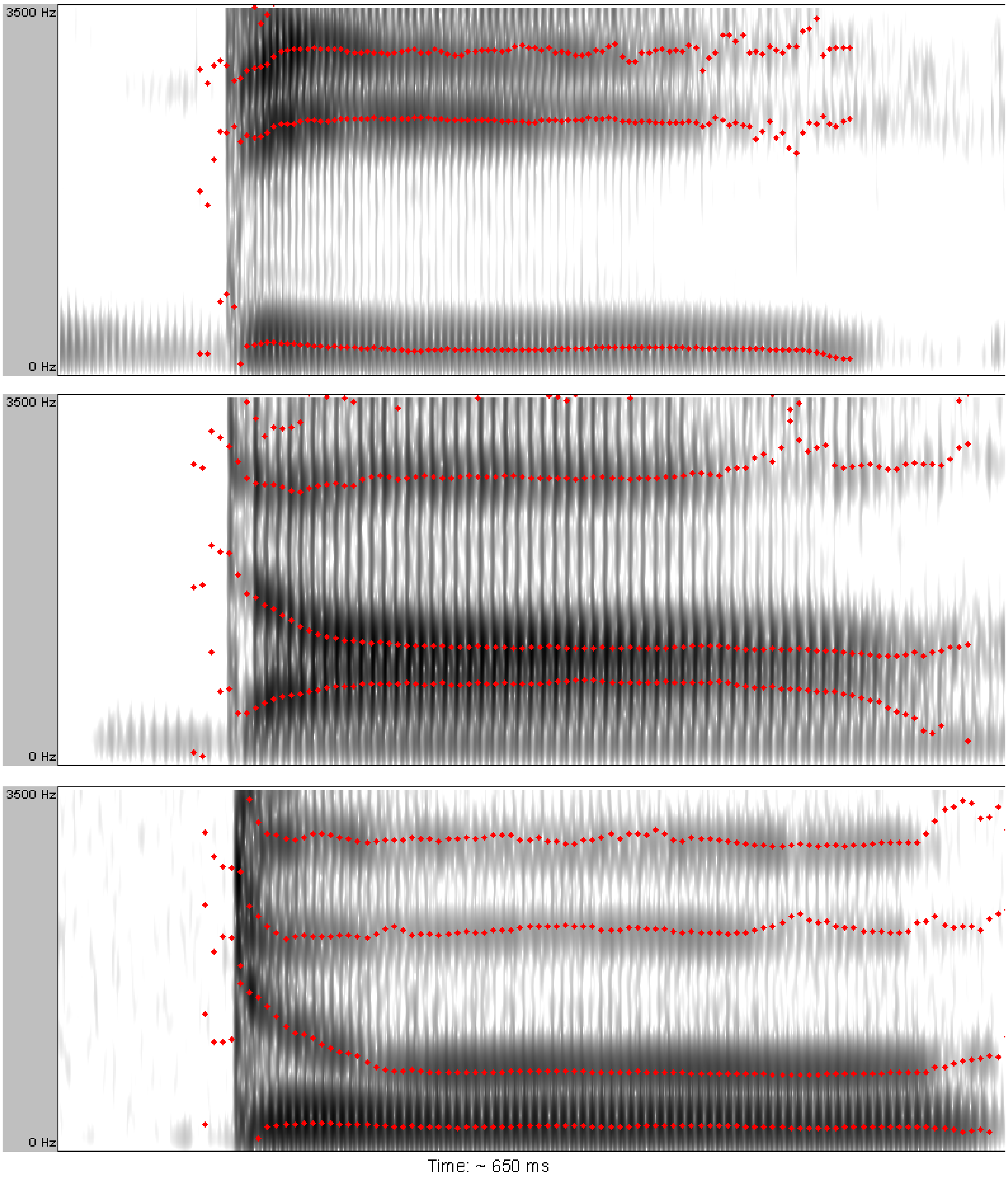
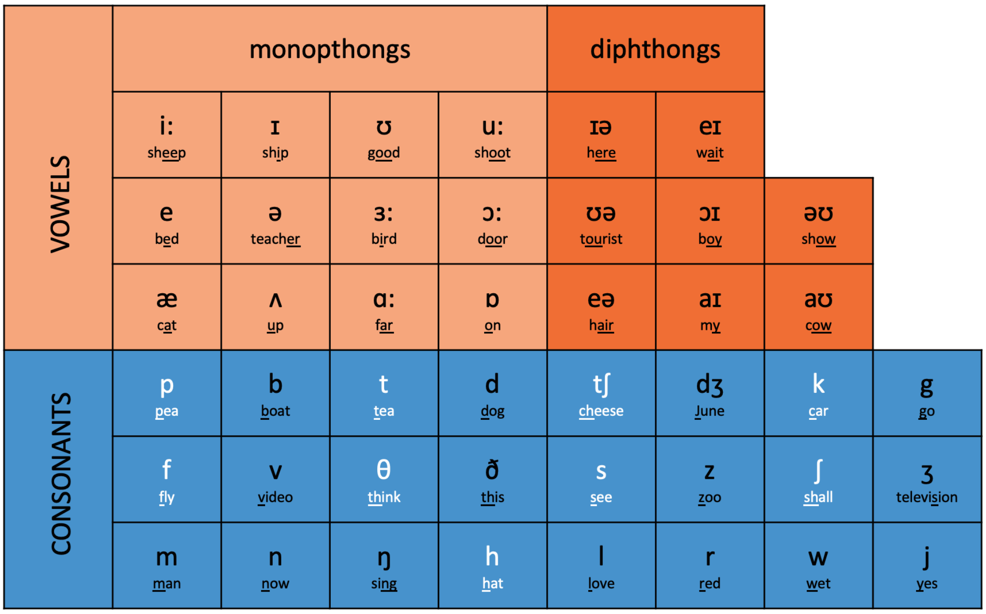


Figure 2: Spectrograms of the syllables “dee”, “daa”, “doo.” spoken by a man. Formants are highlighted in red. Vowels are distinguished from one another by their different formant patterns. Image from [17]

its corresponding acoustic signal is not entirely clear and does not adhere to any consistent rules [21]. This is due to the fact that a single phoneme can possess very different acoustic characteristics depending on the speaker, phonetic context, and prosodic pattern. For example, the duration for which a certain phoneme is pronounced is largely affected by the preceding and successive phonemes, referred to as its pentaphone context [22]. Mood, tone, and emphasis of certain phonemes convey emotions and extra information that cannot easily be modeled acoustically, but can easily be understood by listeners. Simply modeling and analyzing acoustic features of speech signals does not suffice to understand the complex nature of human speech perception. It is clear that humans have a distinct sensitivity for speech sounds that other animals do not possess to the same extent. [18]



	monopthongs				diphthongs			
VOWELS	i: sheep	ɪ ship	ʊ good	u: shoot	ɪə here	eɪ wait		
	e bed	ə teacher	ɜ: bird	ɔ: door	ʊə tourist	ɔɪ boy	əʊ show	
	æ cat	ʌ up	ɑ: far	ɒ on	eə hair	aɪ my	aʊ cow	
CONSONANTS	p pea	b boat	t tea	d dog	tʃ cheese	dʒ June	k car	g go
	f fly	v video	θ think	ð this	s see	z zoo	ʃ shall	ʒ television
	m man	n now	ŋ sing	h hat	l love	r red	w wet	j yes

Figure 3: Chart of phonemes in the English language, including example words within which each phoneme is pronounced. Voiced consonants are in black, while unvoiced consonants are in white.

2.2 Locked in syndrome

Locked in syndrome (LIS) is a rare neurological condition characterized by quadriplegia and inability to speak. A person with LIS is aware and conscious, but cannot move or speak due to complete paralysis of most voluntary muscles, with the exception of eye movements and blinking. LIS results from damage to parts of the lower brain and brainstem, with injuries to the pons being the most common area of damage. Examples of causes include brainstem stroke or lesion, circulatory system diseases, and diseases which destruct the myelin sheath of nerve cells. While vocal cords may not necessarily be paralyzed, LIS interferes with the coordination between breathing and voiced sounds, which prevents people with the syndrome from generating intelligible speech sounds.

Though speech is compromised due to paralysis; linguistic, emotional, and intellectual functions are still intact. This calls for alternative channels of communication. One of the simplest communication methods is via blinking to signify simple responses such as “yes” or “no”. Eye-tracking technology is also often used in communication AT applications. An eye-tracking camera is mounted below a screen that contains cells with different prompts, such as commands or letters. The camera recognizes where the user is looking for a prolonged time and inputs the contents of that cell to the system. [7] Scanning systems allow users to select items from a sequence using a signal such as a sound or blink [9]. The efficiency of speech generation is quite limited using these methods. A recent alternative is to detect imagined speech or intention using electroencephalography (EEG), event-related potentials (ERPs), and even neuroimaging such as functional magnetic resonance imaging (fMRI) and functional near-infrared spectroscopy (fNIRS). Neuroprostheses can decode imagined speech directly from neural signals in the brain. [9] While great strides have been made in the development of AT for communication, the user experience would improve greatly with the addition of a custom voice element.

2.3 Evolution of state of the art in artificial speech synthesis

In the last decade, innovations in artificial speech synthesis based on machine learning paradigms have brought forth tremendous progress in the field of speech technology. In order to appreciate and understand these recent developments, we present a brief review of the developments in the state of the art in TTS systems.

2.3.1 Mechanical speech synthesis

The idea of generating artificial speech has captivated the minds of scientists for centuries. Late medieval scholars in the early 1200s, such as Roger Bacon, are said to have been involved in the creation of “Brazen Heads;” machines with the magical ability to emulate human speech. The first successful mechanical speaking device was developed in the late 1700s by German-Danish scientist Christian Gottlieb-Kratzenstein. He designed acoustic five resonators similar to the human vocal tract which could be activated with vibrating reeds to produce the five long vowels /a/, /e/, /i/, /o/, and /u/. A few years later, Hungarian scientist Wolfgang von Kempelen demonstrated the functionality of his “Acoustical Mechanical Speech Machine,” which could generate single sounds as well as certain combinations of sounds, which resembled words. He created mechanical versions of the components of the human vocal tract; the machine consisted of a pressure chamber to act as the lungs, a vibrating reed as the vocal cords, a rubber tube as the vocal tract, and cavities as the mouth. The machine was operated via a set of controls that adjusted the opening of closing of the cavities. [23]

2.3.2 Concatenative speech synthesis

In concatenative speech synthesis, pre-recorded segments of speech are chained together to form words and sentences. The method can be divided into three main subtypes: unit selection, diphone selection, and domain-specific.

Unit selection utilizes a large database of utterances segmented into individual phones, diphones, syllables, words, phrases, or sentences, or, alternatively, a combination of these. The database may include multiple occurrences of the same unit, possibly with varying pronunciation depending on its pentaphonic context and position within a sentence. Each unit of the segmented database is then given an index based on parameters such as fundamental frequency, duration, position in word or syllable, and its neighboring units. During synthesis, the string of units most compatible with the target sentence is selected from the database with a weighted decision tree. Unit selection synthesized audio often provides the most natural sounding speech, at best indistinguishable from human speech. In order to reach these levels of naturalness, though, a huge database of recorded audio is required, with up to dozens of hours of speech and terabytes of memory. [24] For many text to speech systems, the large amount of data, as well as the computational resources required are not feasible for its application.

Diphone concatenative synthesis generates a waveform by selecting compatible units from a database of sound transitions called diphones. Phones represent the basic, indivisible sounds that occur in a certain language. Diphones thus represent the sound that is made when transitioning from one phone to another. Each language has a limited number of diphones due to phonotactic constraints between phones. To synthesize a target utterance, the input text is first converted into phones and matched with the diphones from the database. Then, digital signal processing methods are performed on the resulting waveform to smooth the transition between diphones and to modify pitch and duration to meet prosodic requirements. An advantage of diphone synthesis is its minimal database, but this also causes synthesized audio to sound robotic and unnatural due to the lack of variation in the database. [25]

Domain specific synthesis builds utterances by selecting units from a small database of whole words and sentences. Because the system is confined to the words and phrases included in the database, its applications are usually limited to a certain domain, such as transit schedules and talking clocks. The system is easy to implement and the generated audio sounds natural, but does not allow for much customization.

2.3.3 Formant speech synthesis

Contrary to concatenative synthesis, formant synthesis does not utilize any pre-recorded speech signals. Instead, artificial speech signals are generated through a spectral shaping system based on acoustic parameters [26]. The method is based on additive synthesis and the source-filter model of speech production. The source-filter model describes speech production as the combination of a source of sound, the vocal cords, and a linear acoustic filter [27]. The goal of formant synthesis is to acoustically simulate the source and filter as accurately as possible. The system is driven by an excitation source, which represents the glottal pulses of voiced sounds, as well as a random noise generator, for unvoiced sounds. Extensive speech analysis is performed to determine a set of rules for control parameters such as fundamental frequency, formants, and voicing source amplitude. Normally analysis is performed at the phoneme level. The source is then passed through a sequence of second-order filters, also called resonators, which reflect the appropriate parameters. [28] The filters can be connected in various ways. In a cascade formant synthesizer, filters are connected in series; the output of each resonator is the input of the next. In this structure formant frequency is the only control parameter required. The cascade structure works well for vowels, but struggles with fricative and plosive consonants. A parallel formant structure involves filters connected in parallel. The excitation pulse is passed through all filters at the same time and their outputs summed. The bandwidth and gain are additional control parameters for each formant. The parallel structure has been found to work better with nasals and fricatives, but not all vowels can be generated accurately. [29] With the advantages and shortcomings of both models in mind, in 1980, Dennis Klatt proposed a solution which incorporated both cascade and parallel structures, with additional parameters for more complex sounds. The model is controlled with 39 parameters updated every 5 milliseconds. [30] The quality of speech generated by the Klatt Formant Synthesizer proved very impressive and paved the way for several other well-known formant synthesizer technologies, such as MITalk, DECtalk, and Klattalk [29].

2.3.4 Articulatory speech synthesis

Articulatory speech synthesis is based on computational models of the physical human vocal tract and the processes which take place there. The system consists of three modules: (1) control model for vocal tract movements, (2) vocal tract model to convert movements into sequence of vocal tract configurations, and (3) acoustic model to convert articulatory information into sound signals. The control model essentially defines the set of articulator movements required to produce speech for each time instance. The vocal tract model generates geometrical information, such as shape and position of articulators, and their movement over time. The vocal tract model can be modeled statistically, biomechanically, or geometrically. The acoustic model calculates the the resulting acoustic speech signal based on digitally simulated air flow and air pressure distribution through the vocal tract model. [31] Articulatory speech synthesis is very closely related to the natural process of human speech, which is an advantage over other synthesis methods [32]. At the same time, though, the highly complex nature of natural speech would require a highly complex model in order to sound as human-like as the speech generated by concatenative synthesis, for example. Articulatory synthesizers have not yet reached this level of naturalness, though much progress has been made. [31]

2.3.5 Statistical parametric speech synthesis

Statistical parametric speech synthesis (SPSS) refers to a series of speech synthesis methods in which the connection between linguistic features and acoustic features are learned via machine learning techniques with a statistical parametric acoustic model [4]. The task of the model is to convert linguistic features of text into acoustic features at a fixed frame rate by using probability mapping. The mapping is learned by training the model on a speech corpus, which contains spoken utterances, along with their corresponding textual and phonetic transcriptions. [14] Before training, a text analysis module is used to extract linguistic features from text. The complex set of features includes a summary of all of the contexts which may affect how each phoneme sounds, including its pentaphonic context, word- and sentence-wise position, and prosodic pattern. [33] Each phoneme in the corpus is labeled with these features, forming context-dependent phonemes [34]. Similarly, acoustic features are extracted from the utterances with a fixed frame rate. Acoustic features describe the shape of the spectral envelope as well as the rate of change of these parameters. [33] During the training phase a statistical parametric model is trained to minimize the error between the extracted acoustic parameters and the corresponding context-dependent phoneme [14]. The model is constructed on the basis of the context-dependent phonemes [33]. A commonly used model is the Hidden Markov model (HMM) because it is particularly useful for modeling sequential data with hidden causal factors [35]. During synthesis, the desired text is converted into context-dependent phonemes in the same way as during the training phase [33]. The possible acoustic features found in a given phoneme are modeled with a Gaussian mixture model (GMM) [14]. The model finds the most likely sequence of acoustic parameters given the sequence of context-dependent phonemes [33]. The produced sequence of acoustic parameters are then fed as input to a waveform generation module to produce the desired speech waveform [14]. SPSS uses statistical methods to generalize a limited set of linguistic features in order to produce speech sounds which it has not encountered before, making it much more flexible than concatenative synthesis. Due to this flexibility SPSS is also language independent and can be conditioned to different voice characteristics.

2.3.6 Deep learning speech synthesis

Recent advances in speech synthesis technology have led to significant improvements in the intelligibility and clarity of artificial speech. The research goal has thus shifted to generating natural and expressive speech, which strongly resembles that of a human. One of the main challenges faced by previous speech synthesis systems is capturing the complex contextual structure of natural language. As previously mentioned, HMM SPSS systems utilize decision trees to cluster complicated context dependent linguistic features and map them onto probability densities of acoustic features. Deep learning (DL) based methods have the ability to directly map linguistic features to acoustic features through deep neural networks (DNN). [4]. DNNs are inspired by the human brain, which learns a new task through labeled examples. Networks contain densely interconnected nodes arranged into hidden layers which apply transformations to the input data. Each node is connected to one or more nodes in the layers below or above. A simplified diagram of a neural network is shown in Fig 4. Typically networks will contain thousands of nodes and various different kinds of connections. [36]

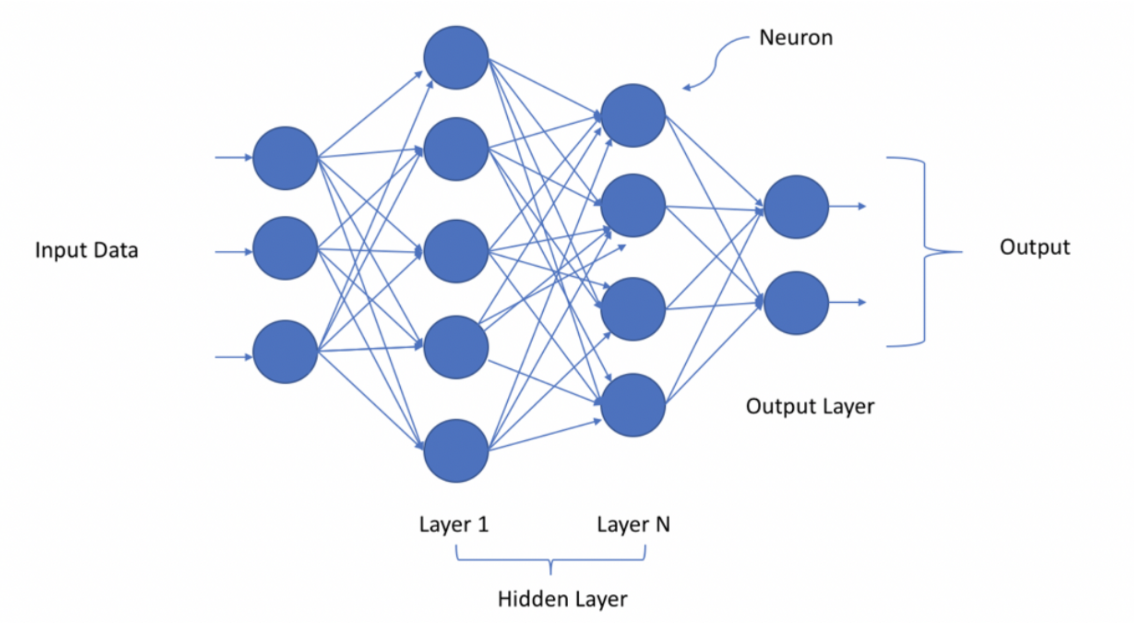


Figure 4: Simple example of connections between nodes in a neural network. Nodes are organized into layers and connected to one another through various transformations. Image from [37]

The output of a single node is a weighted sum of all its inputs with an added bias term. Weight and bias are both learnable parameters within the network, which are adjusted as the network is trained to produce correct output. Prior to training, their values are randomized. The weighted sum is then passed through an activation function to generate the output value of the node. Neural network layers can be organized in a multitude of different ways, depending on the application.

DNNs progressively learn to extract higher-level features from raw input. Unlike in SPSS, these features are not usually known, as their values are located within the hidden layers of the network. A distinguishing feature of DNNs is that the network learns which features belong in which level of representation on its own, eliminating the need for human annotation. This is especially useful in applications involving complex, hierarchical and context-dependent data, such as audio. DNNs have the ability to decipher the hidden internal structures of speech data and employ powerful modeling structures to characterize the feature representations of text and speech. [36]

Similar to SPSS, DL based speech synthesis includes a training phase during which an extensive corpus of text and speech pairs are employed to learn the relationship between linguistic and acoustic features. The TTS pipeline typically includes three components: a text analysis frontend, an acoustic model network, and an audio synthesis network, also called a vocoder. [38]. The pipeline of a typical TTS system is illustrated in figure 5. In the text normalization stage text is converted into a form interpretable by the rest of the network. Abbreviations, numerical values, and other linguistic features are first translated into words. The text is then encoded into a numerical vector form based on a set of rules. The acoustic model component is typically a DNN which transforms normalized text into mel-spectrograms (mel-spectrogram synthesizer). The network is thus trained with pairs comprising of a normalized text segment and its corresponding mel-spectrogram. The vocoder, in turn, generates a waveform conditioned on a mel-spectrogram input. [4] Traditionally each component is trained separately. Recently efforts have been employed into the development of end-to-end TTS systems in which the network is trained directly on text and speech pairs. The method potentially prevents the cumulation of errors and allows for even less manual annotation. [1].

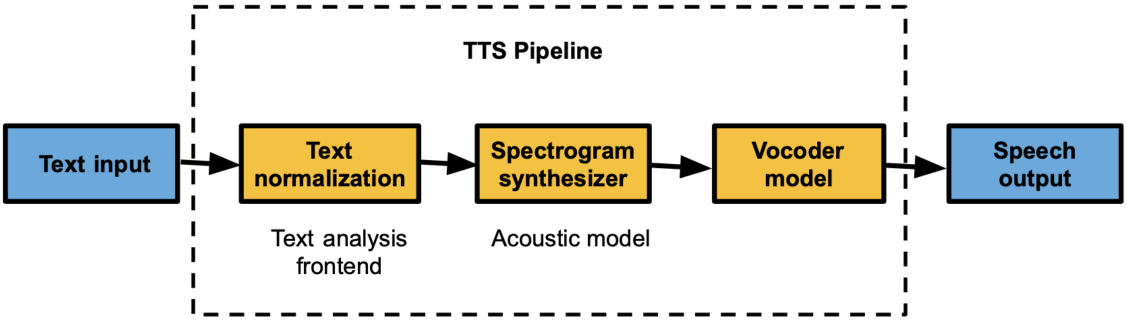


Figure 5: Pipeline of a typical deep learning TTS system. Blue represents input and output, while yellow represents the components within the TTS network. The text analysis frontend normalizes text into vectors. The mel-spectrogram synthesizer is a deep neural network which produces mel-spectrograms conditioned on text input. The vocoder generates audio conditioned on mel-soectrogram frames. Each network is trained with pairs of text and speech.

WaveNet [39] was the first artificial speech model that could successfully model raw audio waveforms instead of acoustic features. The deep convolutional network autoregressively models the probability distribution of a single sample of an audio waveform based on previous samples. WaveNet uses dilated convolutions to increase the receptive field of the network and introduce nonlinearity, which is important when modeling raw audio waveforms. In order to perform TTS, WaveNet must be conditioned on acoustic features. Therefore WaveNet acts as a vocoder in the typical TTS pipeline described previously.

Tacotron [1] is a sequence-to-sequence model which generates a mel-spectrogram from text input alone, without the need for manual alignment. The vocoder is the Griffin-Lim phase reconstruction algorithm. Tacotron follows an encoder-decoder structure with attention. The encoder receives text as input and the decoder produces spectrogram frames to act as acoustic features based on a weighted sum of encoder outputs. The attention mechanism learns the alignment between text input and output mel-spectrogram frames, so there is no need for manual alignment [40].

Tacotron 2 [41] improves the original architecture by replacing the Griffin-Lim algorithm with a modified WaveNet and several other small changes to the original network, such as replacing spectrograms with mel-spectrograms. Tacotron 2 has received a near-human speech rating and is often considered state of the art in TTS systems [4].

Flow based TTS models use the statistical method of normalizing flows to construct a complex unknown distribution from a simple distribution. The model learns an invertible mapping between a latent space and the space which is representative of the training data. By passing samples through the sequence of invertible transformations, we eventually obtain the unknown probability distribution. Training a flow-based model is simple and stable and allows for control over speech styles through manipulation of latent space [6]. Two popular examples of flow-based models are Flowtron [6] and WaveGlow [10], which are described in more detail in the following sections.

3 Models and Algorithms

The approach to voice cloning employed in this thesis is based on the Flowtron mel-spectrogram synthesizer [6] in combination with the WaveGlow vocoder [10]. Flowtron generates a mel-spectrogram conditioned on input text and speaker embedding. Speaker embedding is a representation of the voice characteristics of a speaker in vector form, in which similar speakers are nearer to one another in latent space. WaveGlow infers an audio waveform conditioned on the mel-spectrogram generated by Flowtron. The synthesizer and vocoder are trained separately on pairs consisting of an audio segment and its corresponding text transcription. During inference, Flowtron receives a text input and a speaker ID as input. The speaker ID corresponds to the embedding of the desired speaker. Text input is processed as a vector representation of its phoneme sequence. WaveGlow receives the mel-spectrogram output of Flowtron as input and generates the speech waveform.

3.1 Mel-spectrogram

Most voice cloning methods use mel-spectrograms as an intermediate representation of speech. The reason that Flowtron and most other TTS systems do not directly model waveforms is because audio is a dense domain as well as highly nonlinear. Spectrograms summarize acoustic features in a smoother and simpler way than their waveform counterparts. The two-dimensionality of spectrograms also helps models to learn spatial connectivities between features. Unfortunately, though, a spectrogram representation of audio discards the phase information and there is no unique inverse transformation from spectrogram to audio waveform. Therefore the vocoder component is essential to produce meaningful speech. [4]

A spectrogram shows how magnitudes of frequency components of a signal change over time and it is obtained by taking the STFT of a speech signal. The spectrogram captures audio characteristics such as fundamental frequency (F0), formants, and aperiodicities found in speech. F0, being the lowest frequency of a periodic waveform, is seen as a brighter spectral line at the lowest end of the frequency axis, and the subsequent brighter set of spectral lines represent the formants. Different speech sounds have distinct recognizeable patterns in a spectrogram.

The most important information pertaining to speech tends to be found in the lower frequencies. For this reason, many TTS systems convert frequencies into mel-scale before taking the STFT transform. The mel scale is a perceptual scale of pitches that is judged by humans to be equal in distance from one another. The reference point is 1000 Hz, 40 dB above the listener's threshold, which equates to 1000 mels. The nonlinear transformation from Hertz to mel pitch is calculated with the formula:

$$m = 2595 * log_{10}(1 + \frac{f}{700}), \tag{1}$$

where m = pitch in the mel scale and f is the frequency in Hertz. A plot of mel-scale against the Hertz scale is shown in Figure 6. As can be seen in the plot, at frequencies above 500 Hz, increasingly large distances between frequencies are perceived to have equal pitch intervals. The mel-spectrogram thus emphasizes audio features in the lower frequencies, which are more significant for the analysis of speech.

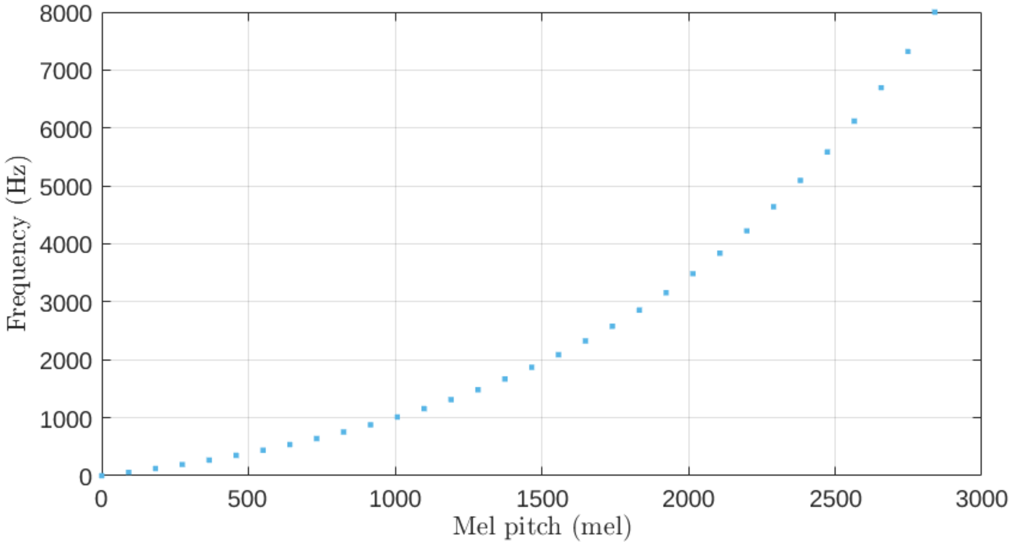


Figure 6: Plot of mel-scale in comparison with the Hertz scale. The mel-scale of pitches are perceived by humans as equal in distance from one another.

Figure 7 shows the mel-spectrogram of an audio recording. The speech formant patterns in the lower frequencies are more clearly distinguishable from the rest of the spectrogram.

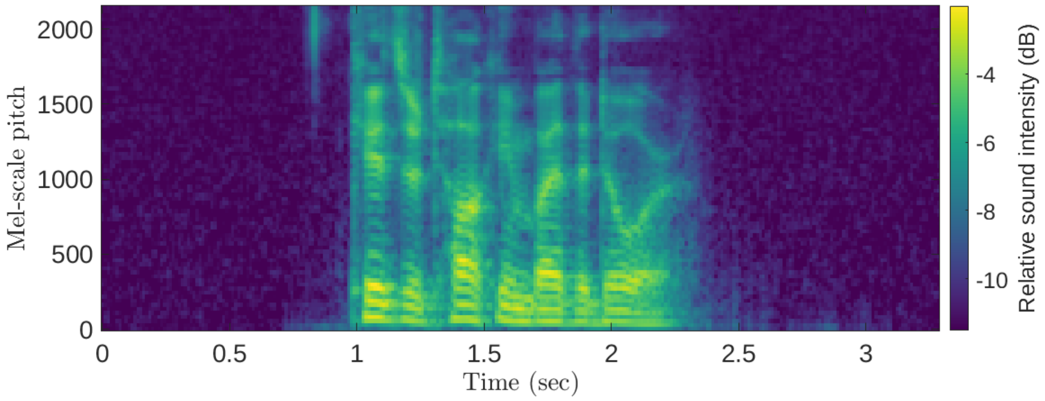


Figure 7: Mel-spectrogram of a man saying “Kids are talking by the door.” The mel-scale is a more accurate representation of how sounds are judged by humans.

3.2 Normalizing Flows

Flowtron and WaveGlow are both based on the statistical method of normalizing flows for probability density estimation. The idea behind the method is to reconstruct a known, simple distribution $p(z)$ into a more complex one $p(x)$ by passing it through a chain of invertible transformations, called steps of flow. We sample a latent variable z and pass it through K steps of flow in order to produce a sample x from the target distribution $p(x)$. Each transformation f must be bijective, which means the following must be true:

$$x = f(z), z = f^{-1}(x) \tag{2}$$

x is produced through the steps of flow:

$$x = f_1 \circ f_2 \circ ... \circ f_k(z) \tag{3}$$

Since the steps of flow are invertible, the following must be true:

$$z = f_k^{-1} \circ f_{k-1}^{-1} \circ ... \circ f_1^{-1}(x) \tag{4}$$

We can estimate the unknown probability distribution for one step of flow using the Change of Variables rule as follows:

$$p(x) = p(z) \mid det(J(f^{-1}(x))) \mid \tag{5}$$

where J is the Jacobian of the inverse transform $f_i^{-1}(x)$. Therefore we can evaluate the negative log likelihood of the target distribution $p(x)$ for one step of flow as a cost function:

$$logp(x) = logp(z) + log \mid det(J(f^{-1}(x)) \mid) \tag{6}$$

The same idea can be expanded to multiple steps of flow as follows:

$$logp(x) = logp(z) + \sum_{i=1}^{k} log \mid det(J(f_i^{-1}(x)) \mid) \tag{7}$$

3.2.1 Flowtron

Flowtron is an autoregressive flow which produces a sequence of mel-spectrogram frames conditioned on text and speaker embeddings. We define the known distribution as a zero-mean spherical Gaussian $z \sim N(z; 0, \sigma^2)$ and apply a series of transformations f to produce a sample from the target distribution $p(x)$, which represents mel-spectrogram space. The latent variable z has the same number of dimensions and frames as the resulting mel-spectrogram sample. The mapping between the latent space and mel-spectrogram space is determined through Masked Autoregressive Flow (MAF), which is an implementation of normalizing flows where the transformation layer is constructed using affine coupling layers. The key behind affine coupling is to choose the invertible transformations in such a way that the functions and their log determinants are computationally efficient. [42] Each successive mel-spectrogram frame depends on previous frames. The previous frames z_{t-1} produce scale and bias terms, s_t and b_t respectively, which affine-transform the succeeding frame z_t:

$$(logs_t, b_t) = NN(z_{1:t-1}, text, speaker) \tag{8}$$

$NN()$ represents an autoregressive causal transformation, which in this case, is a neural network. *text* and *speaker* represent the text and speaker embeddings which are concatenated to the sample during training and inference. While the coupling layers must be invertible, $NN()$ does not need to be. The scale and bias operations can be arbitrarily complex operations because the coupling layer itself preserves the invertibility for the overall network. This is true because s_t and b_t only depend on previous frames $z_{1:t-1}$ and fixed speaker and text vectors. [6] We can then calculate the forward and inverse steps of flow as follows.

$$f(z_t) = (z_t - b_t) \circ s_t \tag{9}$$

$$f^{-1}(z_t) = s_t \circ z_t + b_t \tag{10}$$

The resulting frame z_t can then simply be concatenated to form the new input vector for the next iteration:

$$z_{1:t} = concat(z_{1:t-1}, z_t) \tag{11}$$

The Jacobian of the coupling layer can be easily computed as follows:

$$\frac{\partial f(z_t)}{\partial z^T} = \begin{bmatrix} I_{t-1} & 0 \\ \frac{\partial f(z_t)}{\partial z_{1:t-1}} & diag(s \odot z_{1:1-t}) \end{bmatrix} \tag{12}$$

where I_{t-1} is an identity matrix and $\text{diag}(s \odot z_{1:1-t})$ is a diagonal matrix with the scaling values. The matrix is lower triangular, which means that its determinant is simply the product of its diagonal values:

$$det(\frac{\partial f}{\partial x^T}) = \sum_i s_i \tag{13}$$

Therefore we can compute the cost function with the determinant from the coupling layer:

$$p(x) = logp(z) + \sum_{i=1}^{k} log \mid s_i \mid \tag{14}$$

As can be seen, only the s terms affect the probability density mapping. The coupling layers thus remain tractable and the cost function efficient to compute when the transformation functions are chosen in such a way that they form a triangular Jacobian [43].

The flowchart in Figure 8 illustrates the Flowtron concept visually. Yellow blocks represent the training phase and blue blocks the inference phase. During training, the Flowtron model technically learns the inverse transformation $f^{-1}(x) = x'$. After K steps of flow, the network learns to transform x into a sample from $p(z)$. After training, we sample z from $p(z)$ and perform the forward transformation $f(z) = z'$. After K steps of flow we produce a sample from the target distribution $p(x)$.

3.2.2 WaveGlow

A pretrained WaveGlow model[1] provides the vocoder component. WaveGlow adopts insights from Glow [44] and WaveNet [39]. WaveGlow, though, is not autoregressive, leading to fast audio synthesis in comparison to other models.

WaveGlow works on a similar principle as Flowtron; the network learns an invertible mapping of data to a latent spherical Gaussian space with bijective affine coupling layers. The model is also trained by directly minimizing the NLL of the audio training data, which can be calculated using the Change of Variables theorem, as in equations 5 to 7. In this case, we model the distribution of audio samples conditioned on mel-spectrograms. [10] WaveGlow, though, uses a second kind of bijective transformation in addition to affine coupling layers. Batches of 8 audio samples are converted into vector form and processed through the network in the forward pass. One step of flow in WaveGlow consists of an invertible 1x1 convolution, followed by the familiar affine coupling layer. Half of the input audio vectors x are left unchanged and produce the s and b terms used to scale and translate the rest of the input, similar to equation 8:

$$x_a, x_b = split(x) \tag{15}$$

[1] https://github.com/NVIDIA/waveglow

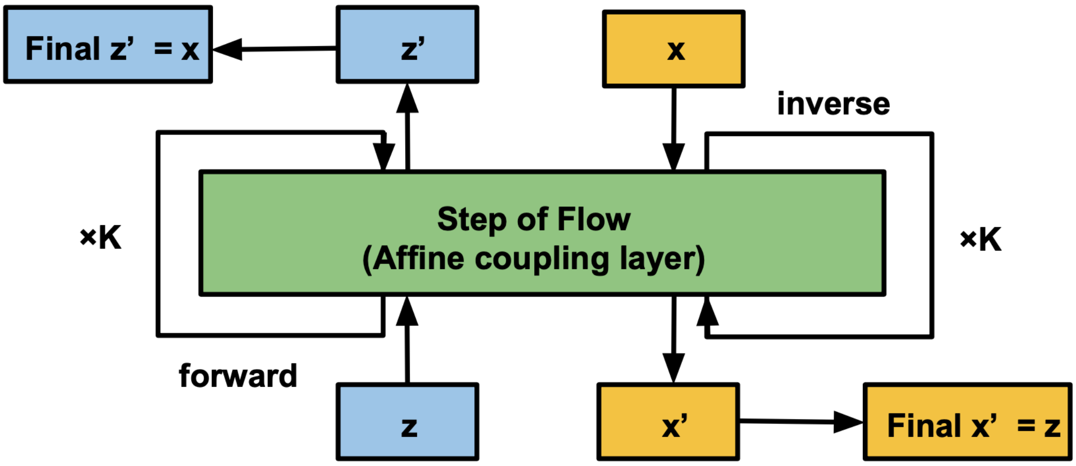


Figure 8: Flowchart of the Flowtron mel-spectrogram synthesis pipeline, which autoregressively produces mel-spectrogram frames conditioned on text and speaker embedding. Blue blocks represent the inference phase, while yellow the training phase. The green block represents a single invertible transformation known as a step of flow. The scale and bias terms of the transformations are determined through affine coupling layers. During training, a sample x from the mel-spectrogram space $p(x)$ is transformed into a latent variable z from a zero-mean spherical Gaussian $p(z)$ through K transformations of $f^{-1}(x)$. During inference, we invert each f^{-1} to form the forward transformation f. Then we transform z into x by running it through $f(z)$ for K steps of flow.

$$(logs, b) = WN(x_a, melspectrogram) \tag{16}$$

$$x'_b = s \circ x_b + b \tag{17}$$

$$f^{-1}_{coupling}(x) = concat(x_a, x'_b) \tag{18}$$

$WN()$ is a transformation consisting of dilated convolutions similar to WaveNet [39], with the exception that WaveGlow has noncausal convolutions. The corresponding upsampled mel-spectrogram is added in the affine coupling layer in order to condition the generated result, as written in equation 16.

Information from the same half of input do not directly modify one another in the affine coupling layer. Following the method used in Glow [44] a 1x1 affine coupling layer is added before each affine coupling layer in order to maximize the information gained during training. The weights W of the 1x1 convolutions are initialized to be orthonormal in order to preserve invertibility. The log-determinant of the Jacobian of the transformation can simply be added to the loss function, due to the Change of Variables theorem.

$$f_{conv}^{-1} = Wx \tag{19}$$

$$log|det(J(f_{conv}^{-1}(x)))| = log|detW| \tag{20}$$

After adding together the terms from the couplings layers the full loss function is calculated as follows:

$$logp_{\theta}(x) = -\frac{z(x)^T z(x)}{2\sigma^2} \tag{21}$$

$$+ \sum_{j=0}^{n-coupling} logs_j(x, mel - spectrogram) \tag{22}$$

$$+ \sum_{k=0}^{n-conv} logdet|W_k| \tag{23}$$

The first term (Eq. 21) is the log-likelihood of the spherical Gaussian, where σ^2 is the variance of the distribution. Terms 22 and 23 arise from the change of variables of the coupling and convolution layers.

3.3 Model architecture

Flowtron consists of a text analysis frontend and mel-spectrogram generator DNN. Similar to Tacotron [1][41] the DNN consists of an encoder-decoder structure with attention. The flowchart in Figure 9 illustrates the Flowtron network in more detail. Each character in the input text is mapped to a text embedding which serves as a vector representation of that character. Each character in the input text, including punctuation, is given a specific numerical value, called a text token. During training text embeddings modify themselves so as to better represent their phonetic characteristics.

The text embeddings are first passed through three one-dimensional convolutional banks with a kernel size of 5 and 512 filters. The output is rescaled and recentered with batch normalization in order to keep training stable.

The output is passed through a bidirectional long short-term memory (LSTM) network with a hidden state of size 256. A LSTM network is a type of recurrent neural network (RNN). RNNs are useful for modeling sequential data like audio and

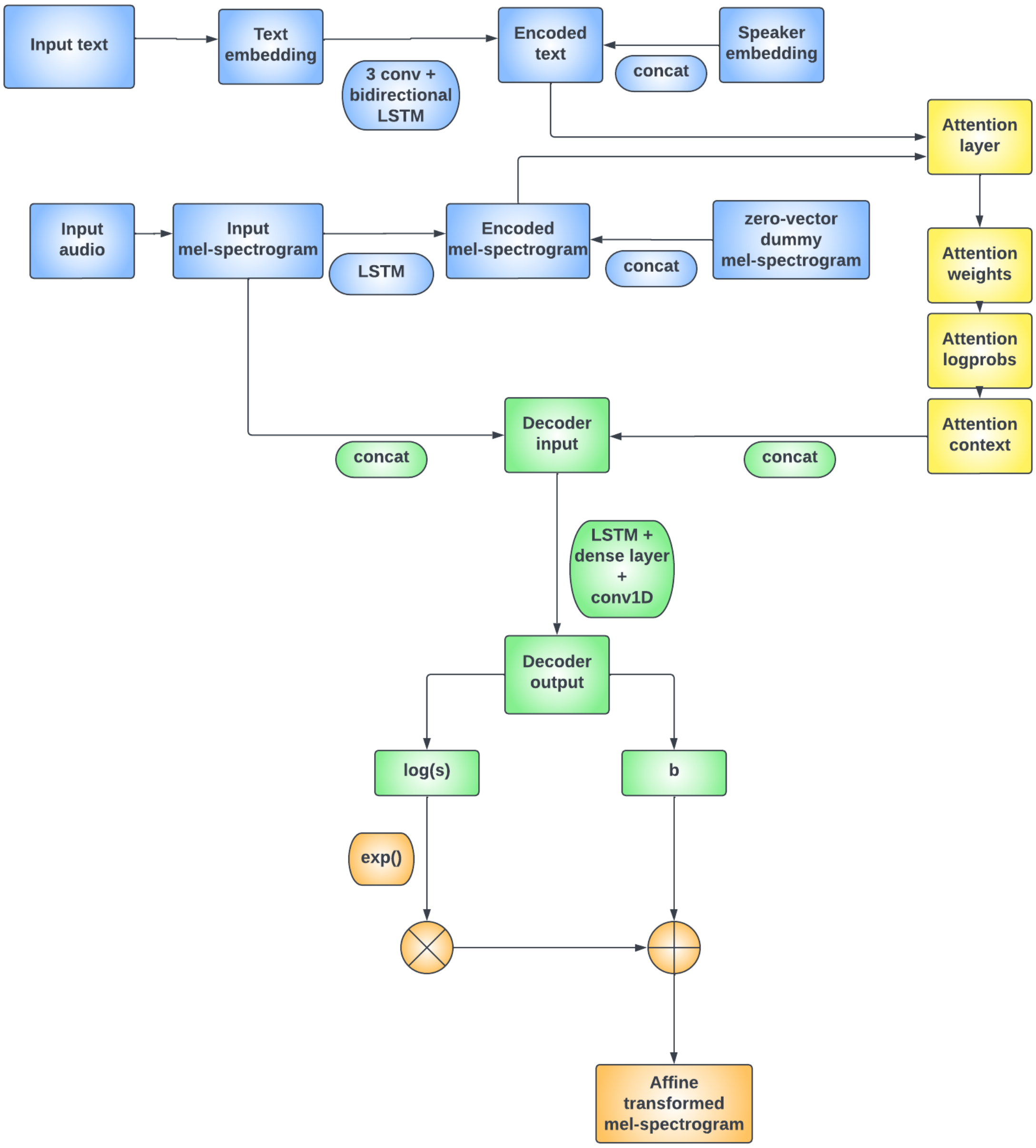


Figure 9: A detailed flowchart of the Flowtron network, which includes all of the network layers. Blue components constitute the encoder module and green components the decoder module. Yellow represents the attention layer and orange the final affine transformation. Square components represent vectors, while oval components represent calculations within network layers.

text data because they maintain previous information in a hidden state, which is analogous to short-term memory. In practice it is difficult to train a standard RNN to solve a task that requires long-term temporal dependencies because the gradient of the loss function tends to decay with time (vanishing gradient problem). LSTM aims to solve this problem by introducing additional memory cells with a set of gates that control when information enters the memory cell, when it is output, and when it is forgotten. The hidden state size refers to number of features expected in the hidden state of the LSTM layer.

An optional fixed speaker embedding of size 128 is concatenated to the end of the text embedding at each token. In a multispeaker model the speaker embeddings condition the network to adopt the unique speech characteristics of the speaker.

The mel-spectrogram of the corresponding utterance is obtained using librosa. A zero-vector is concatenated to the beginning of the spectrogram to serve as the first time step. It is then passed through an LSTM layer with a hidden state size of 1024.

Between the encoder and decoder an attention mechanism is implemented over the encoder hidden states. The attention mechanism in deep learning networks is analogous to cognitive attention. More attention is assigned to parts of the input which are relevant to the output, while less important information is assigned less. The aim of the attention mechanism is to align text tokens (represented by encoder hidden states) to their corresponding mel-spectrogram frames (represented by decoder hidden states) such that resulting speech sounds as natural as possible. The task is not trivial, as alignments are not one-to-one. Text tokens are pronounced for a longer or shorter time, depending on its context, and thus require more or less frames. For each decoder step, all previous encoder hidden states are summarized and scored by their individual relative importance to that decoder step. These scores are known as attention weights. Flowtron uses a content-based tanh attention from [45]. Given a sequence of encoder hidden states $(h_1, ..., h_{T_A})$, and a sequence of decoder inputs $(d_1, ..., d_{T_B})$, the attention weights vector is computed as follows:

$$a_{t,i} = softmax(v^T tanh(W_1' h_i + W_2' d_t) \tag{24}$$

where the vector v and the matrices W_1' and W_2' are learnable parameters of the model. $softmax$ is an activation function which converts a vector of values into a vector of probabilities. Attention weights can be visualized as a matrix between encoder hidden states and decoder hidden states, as seen in figure 10. Ideally, attention weights are assigned such that a diagonal matrix is formed; Encoder timesteps from the beginning, middle, and end of the text input should align with the first, middle, and last decoder timesteps.

From the attention weight vector the context vector for time step t is calculated as follows:

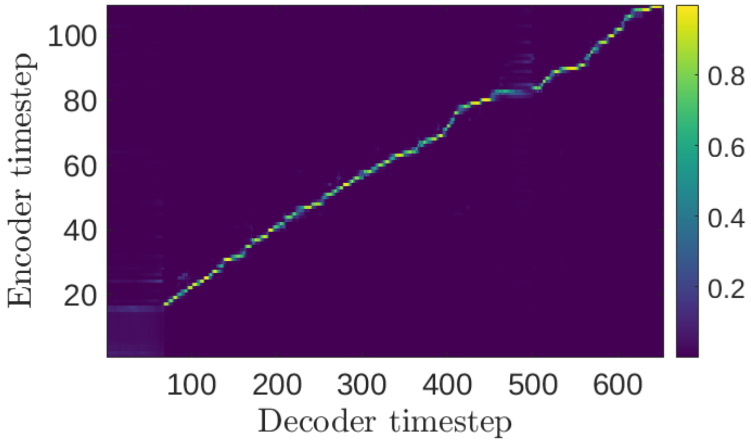


Figure 10: Example of an attention weights matrix with good alignment between text tokens and mel-spectrogram frames. Encoder timesteps represent the text token hidden state representations, while decoder timesteps the mel-spectrogram frame hidden states. The matrix values represent the attention assigned to the corresponding encoder and decoder timesteps. Good alignment is attained when the diagonal values of the matrix are highest.

$$c_t = \sum_{i=1}^{T_A} a_{t,i} h_i \tag{25}$$

The attention context vector is time-wise concatenated to the mel-spectrogram hidden state vector and used as input for the decoder module. This vector is passed through an LSTM with a hidden state size of 1024. The output is passed through a dense linear transformation layer and one-dimensional convolutional layer with 160 filters and a kernel size of 1. The log of the scale term $\log(s)$ constitutes the first half of the output of the convolutional layer and the bias term b the second. All that is left is to affine transform the original input mel-spectrogram with the resulting scale and bias terms and evaluate its log-likelihood.

Once the model has been trained, mel-spectrogram synthesis is simply a matter of sampling from the prior latent distribution $z \sim N(0, \sigma^2)$ and passing it through the trained network in a forward pass. The simple and stable nature of Flowtron allows for control over speech variation and style transfer. Prosody and speech expressivity can be modified by manipulating the latent space before sampling.

The WaveGlow network includes 12 coupling layers and 12 invertible 1x1 convolutions. Each of the coupling layer networks ($WN()$) has 8 layers of dilated convolutions with 512 channels used as residual connections and 256 channels in the skip connections. Two of the channels are output after every four coupling layers. Doing so helps the network to include information from several time scales and for gradients to propagate to previous layers. [10] The authors of [6] hypothesize that WaveGlow can be considered a universal vocoder, meaning that it does not have to be trained with the same data as the Flowtron part of the model. We verify this claim by using a pretrained WaveGlow model during the experimental section and thus focus most of our efforts on training the Flowtron part of model.

3.3.1 Fine-tuning for few-shot speech generation

Voice clone success is highly data-dependent. In cases where available data is not of sufficient size or quality, a pretrained model trained on different data can be fine-tuned to incorporate the voice characteristics of a previously unseen speaker, even with limited data. Flowtron models are fine-tuned in a few-shot setting, in which the model learns to generalize over a small amount of previously unseen data by building upon its existing knowledge. Instead of initializing weights randomly, weights from the pretrained model are transferred to the fine-tuned model. The network layers are frozen, excluding the speaker embedding layer. Then, the model is trained with the unseen speaker data. During training the speaker embedding layer modifies itself to learn voice characteristics of the new target speaker with minimal influence to the rest of the network. General acoustic features, such as pronunciation and prosody, are retained from the pretrained model. Fine-tuning requires less computational power and tends to converge faster than models trained from scratch with randomly initialized weights.

3.3.2 Style transfer for emotional speech synthesis

The style transfer approach is explored with the theme of emotional speech. People express different emotions with the way that they speak. Therefore the possibility to convey how one is feeling through voice characteristics is an important addition to an artificial speech synthesis system. The generation of emotional speech is not simply a matter of altering pitch or speed of speech, which makes it difficult to manually add emotional characteristics to a Flowtron model.

In the style transfer approach, we sample from a posterior distribution conditioned on prior evidence which contain speech characteristics of interest. Prior evidence z_e is collected by performing a forward pass through the trained network with the target voice speaker ID, and text-audio pairs of a set of samples that embody the desired emotion. With this approach, previously unseen accents, prosodies, and other speech styles can be transferred to the target voice without needing to be retrained.

4 Methodology

The goal of the experimental part of the thesis is to train a Flowtron model with the voice of a person living with LIS by using the fine-tuning for few shot speech synthesis paradigm, outlined in section 3.3.1. Control of speech expressiveness and style transfer is also explored. We then showcase speech generated by the model with a simple graphical user interface (GUI). The Flowtron[2] and WaveGlow[3] frameworks are publicly available on GitHub and used with minimal modifications to the code. The training data used in the experiments includes about 15 minutes of relatively noiseless audio from a single speaker. The data is used to fine-tune the speaker embedding layer of various Flowtron models pretrained on professionally recorded speech datasets. For the sake of comparison, we also train a Flowtron model from scratch on our training data, though it is hypothesized that the dataset is not sufficient both quantitatively and qualitatively to successfully clone the target speaker.

4.1 Data preprocessing and required tools

The original target speaker data available consists of a YouTube video with a professor speaking about art in English for a duration of about 15 minutes. The video is first converted to waveform audio file format (wav). Then, using the audio editing software Audacity, the channel format of the file is changed from stereophonic to monophonic and the the sampling rate converted to 22050. The audio file is then divided into short utterances ranging from 1 second to 10 seconds. The division is based on natural stopping points in the speech, so that each utterance constitutes a full sentence. Filler words, such as “umm” and non-speech sounds, such as deep breathing and music, are clipped from the audio. The utterances are transcribed manually, with punctuation included at the end. The final target speaker dataset consists of 127 utterances with a total length of 13 minutes and 26 seconds. The transcriptions are written line by line into a text file, along with the location of the corresponding audio file in the file structure. The resulting text-speech pairs are randomly divided into training and validation sets, with 111 and 16 pairs in each, respectively.

For the emotional style transfer experiments we used the open-source Emotional Voices Database from [46]. It includes utterances from audiobooks spoken by voice actors in four different emotional styles: angry, amused, disgusted, and sleepy. We chose ten utterances from each emotion class spoken by a female speaker labeled “Spk-Bea” to use as prior evidence z_e. We ensured that each utterance was between 1-10 seconds and performed the same audio preprocessing steps as for the target speaker dataset.

[2] https://github.com/NVIDIA/flowtron

[3] https://github.com/NVIDIA/waveglow

The base model for the fine-tune paradigm is a Flowtron multispeaker model[4] trained on the LibriSpeech corpus [47], which includes about 1000 hours of speech from 2300 different speakers reading passages from various audiobooks. All of the speech is recorded in a professional environment but the overall quality of audio differs between speakers. The model is chosen due to its versatility. The speech corpus it is trained on has a rich vocabulary, which provides the resulting model with an extensive pool of examples to learn correct pronunciation from. The variation in audio quality also leads to a model that is more robust to noise. We used a WaveGlow model[5] pretrained on the same LibriSpeech dataset as the vocoder component. WaveGlow is considered a universal vocoder [6], so therefore the model is used without modifications or fine-tuning.

Deep learning models require significant computational power. Therefore access to NVIDIA's Volta or Pascal series GPU is required to train the models. Aalto University's Triton kernel makes it possible to employ the needed power. In addition, the NVIDIA CUDA Deep Neural Network (cuDNN) library of primitives is needed. The Flowtron and WaveGlow architectures run on the PyTorch machine learning framework. The full list of technical prerequisites and detailed steps for Flowtron+WaveGlow setup are outlined on the Flowtron GitHub page.

4.2 Fine-tuning a Flowtron model

Speaker similarity. Each speaker in the Flowtron base model has a unique speaker ID. When setting up the dataset for fine-tuning, the new speaker can either be labeled with a new speaker ID or the ID from a speaker in the base model dataset. The former method introduces a new speaker into the dataset, while the latter fine-tunes the existing speaker directly. While both methods work, we found that fine-tuning an existing speaker which has similar voice characteristics to the target speaker leads to better results.

We could only find information for a subset of 123 speaker IDs out of the 2300 speakers in the base model. This subset constitutes recordings of higher quality and longer duration. In order to determine the speaker most similar to our target speaker we used the Resemblyzer[6] python package. Resemblyzer derives a high-level representation of a voice based on a deep learning model trained on speaker verification. Speaker verification refers to the process of verifying whether an utterance belongs to a certain speaker. The model takes an audio file as input and creates an embedding which summarizes its voice characteristics in vector form. Voices which are similar are close together in latent space, while voices that are very different from each other are further apart. [4]

Ten utterances from each speaker in the base model dataset as well as ten utterances

[4] https://github.com/NVIDIA/flowtron

[5] https://github.com/NVIDIA/waveglow

[6] https://github.com/resemble-ai/Resemblyzer

from our target speaker dataset were fed as input to the model. As output the model produces normalized speaker embeddings for each speaker. Then we calculated the dot product of the target speaker embedding and each of the base model speaker embeddings because the dot product of two normalized embeddings defines their cosine distance. The minimum distance was found for Speaker 669, which we subsequently identified as most similar to the target speaker. The scatter plot shown in Figure 11 illustrates the two-dimensional projections of the speaker embeddings. The dimensionality reduction method used to calculate the projections was Uniform Manifold Approximation and Projection (UMAP). As can be seen, male and female speakers each form a distinct cluster. The target speaker and Speaker 669 are labeled in the plot.

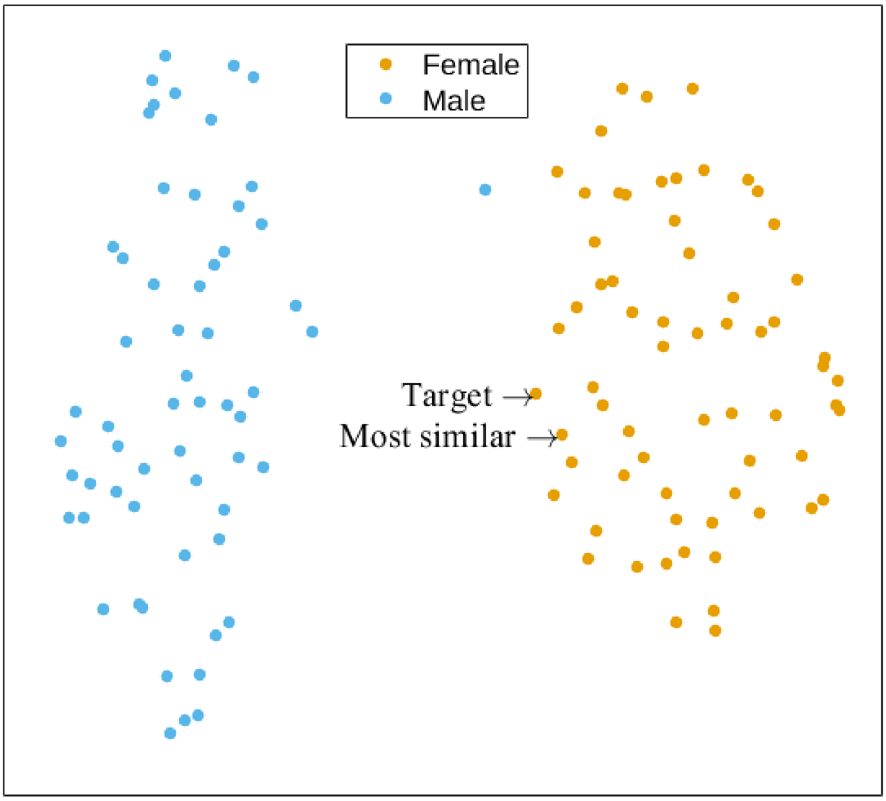


Figure 11: Visualization of the two dimensional projections of 124 speaker embeddings, which are calculated with the Resemblyzer [4] deep learning network trained on speaker verification. The network is able to differentiate between male and female speakers. The target speaker and the speaker with the most similar characteristics both belong in the female speaker cluster.

Training parameters. The success factor of of deep learning applications is largely dependent on the combination of training parameters in the setup. A significant portion of the experimental phase of this thesis involved experimenting with different training setup configurations and evaluating their effect on speech samples.

Three of the most important training parameters in machine learning include batch size, learning rate, and optimization algorithm. Batch size dictates the number of samples processed through the network before the model weights are updated. With a batch size larger than one, an average loss across the samples in the batch is computed, instead of for a single input. Average loss tends to be less noisy because training is less affected by single inputs, some of which may be outliers and as such not representative of the entire dataset. A small batch size converges slower, because of the noisiness. On the other hand, batch sizes that are too large are computationally expensive. In addition the average loss may not change much across batches, which may lead to suboptimal solutions. Learning rate determines the size of the step at which weights are updated at each iteration across a batch. There is a tradeoff between rate of convergence and overshooting. A learning rate that is too high may jump over optimal solutions, while one that is too low may get stuck in a local minima or take too long to converge. The optimization algorithm, in turn, is responsible for finding the value of weights that minimize the error when mapping inputs to outputs.

The parameter σ^2 defines the variance of the latent variable z sampled from the $p(z)$ during training. The lower the value of σ^2, the closer together z values are in latent space. As a result the model is more biased toward the dataset and generated speech is less varied. According to [6], Flowtron produces the best results when training is conducted with $\sigma^2 = 1$ and inference is performed with $\sigma^2 < 1$.

We reduced the learning rate from the default 0.001 to 0.0001 and set the batch size to 16 to optimize training efficiency and stability. Otherwise default training parameters were used. Important training parameters and their values are listed in Table 1.

Table 1: Training parameters used to train a fine-tuned model.

Parameter	Value
Learning rate	0.0001
Batch size	16
Optimization algorithm	RAdam
σ^2	1.0
Speaker ID	669

Mel-spectrogram parameters are also important when it comes to accurate training. Since the STFT provides time-localized information of how the frequency components change over time, varying the window and hop length affects the time and frequency resolution of the resulting signal. We used the default mel-spectrogram parameters described in [6]. The STFT is applied with a fast Fourier transform (FFT) size of 1024, a window size of 1024 samples and a hop size of 256 samples, which corresponds to about 12 ms. The maximum frequency is set to 8000 Hz. The frequency spectrum in mel scale is divided into 80 evenly spaced channels with librosa mel filter banks, which project FFT channels onto mel-frequency channels with a linear transformation matrix. The space between each channel signifies differences in pitch as perceived by humans, instead of the actual distance in the frequency dimension. Mel-spectrogram parameters used in training are summarized in Table 2.

Parameter	Value
Sampling rate	22050
Filter size (FFT)	1024
Window size (samples)	1024
Hop size (samples)	256
Maximum frequency (Hz)	8000
Mel channels	80

Table 2: Mel-spectrogram parameters

Figure 12 shows the NLL training and validation loss progression during training, smoothed with a moving median filter of size 5. A step refers to one iteration across a batch. In the beginning, both training and validation loss decrease at a stable rate, which indicates that the model is learning. Both losses begin to plateau at around Step 10000. Training loss converges at around -1.04, while validation converges at -1.06. The model was trained for a total of 13000 steps, which corresponds to 1625 epochs.

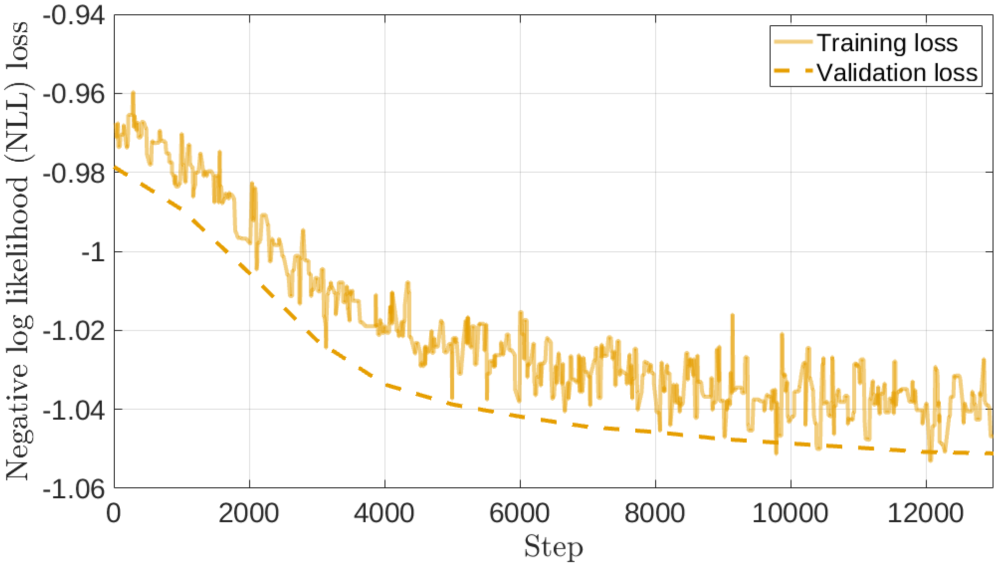


Figure 12: Negative log likelihood (NLL) training and validation loss during training of a fine-tuned Flowtron [6] model. Both validation and training loss decrease at a stable rate and begin to plateau around its 10000th iteration across the network.

4.3 Training a Flowtron model from scratch

In order to verify the necessity of using the fine-tuning method for our dataset, we also trained a Flowtron model from scratch. Instead of transferring weights from a base model and freezing layers, weights were initialized randomly and the model trained across the entire network. We used a batch size of 1 and learning rate of 0.001 to match the parameters used in the Flowtron paper [6]. Training parameters are summarized in Table 3. We used the same mel-spectrogram parameters presented in section 4.2.

Table 3: Training parameters used while training a Flowtron model from scratch according to [6]

Parameter	Value
Learning rate	0.001
Batch size	1
Optimization algorithm	RAdam
σ	1.0

Figure 13 shows the progression of NLL training and validation loss when training from scratch. Validation loss is very high in the beginning, likely due to randomly intialized weights, but it also decreases quickly and plateaus sharply already at step 5000. The training loss curve remains flat over the course of training, which indicates underfitting; the model did not have the capacity to generalize patterns in the training data due to an insufficient amount of examples in the dataset. The losses also remain high in comparison to the fine-tuned model. The model was trained until 100000 Steps, which is equal to 100000 epochs, but no substantial changes in losses occurred after 5000 steps.

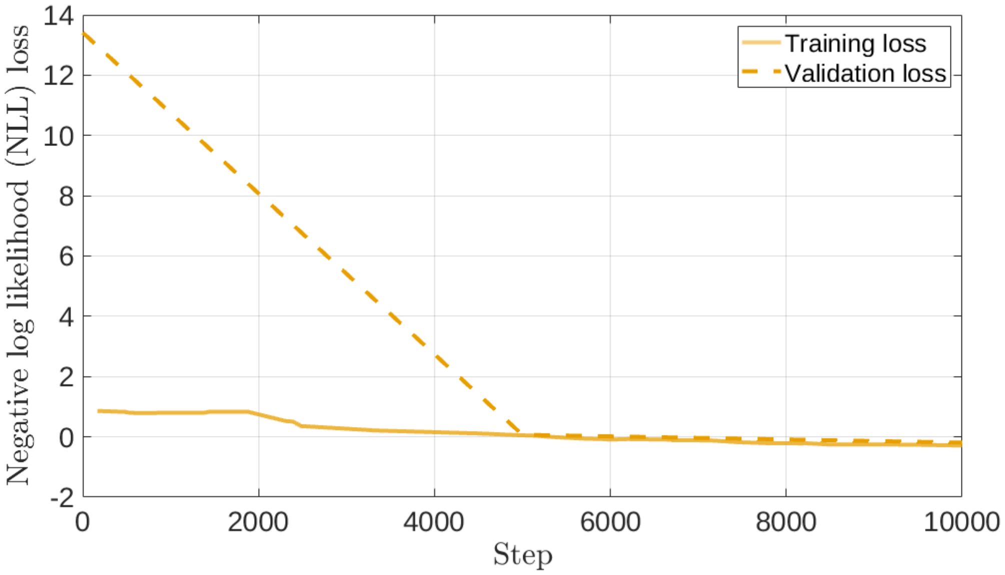


Figure 13: Negative log likelihood (NLL) training and validation loss progression during training of a Flowtron model using limited data. Validation loss falls sharply and plateaus, while training loss remains stagnant. The loss curves indicate underfitting due to insufficient data.

5 Results

In this section we present mel-spectrograms and speech samples synthesized by the fine-tuned model and the model trained from scratch. We conducted a small-scale Mean Opinion Score (MOS) listening test to evaluate the speech from a potential user's perspective. We then present a user-friendly speech synthesis application which people with speech impairments can use as a tool for synthesizing speech.

5.1 Mel-Spectrogram synthesis with Flowtron models

After training mel-spectrogram inference of input text is carried out by randomly sampling z values from a spherical Gaussian and running them through the trained network in the forward direction. During inference we use a variance of $\sigma^2 = 0.5$ unless otherwise specified. In section 5.3 we analyze the effect of different σ^2 values on generated speech.

Four different text prompts are used as input to perform inference from the two trained Flowtron models. Two of the prompts are included in the training data, so the model has encountered them before, while the other two consist of unseen text. The text prompts also vary in length in order to evaluate the model's ability to generate audio of different durations. The prompts are listed in Table 4.

Table 4: Text prompts fed as input to the Flowtron voice clone model. The prompts include text that the model has encountered during training (seen) and new text (unseen). The prompts also vary in length (long and short).

Seen short	Streamlining allowed people to see something very far down in the future and hope and aim for that.
Unseen short	The human voice is the most beautiful instrument of all, but the most difficult to play.
Seen long	I like to make things, so actually I am probably a designer myself, in the deepest of my hearts. But I also like to dream up things, to write about things, and to make stories about things and so this is probably why I became a design historian. Good design is always of its time because it represents what people like at this very moment, what people care about, what people would like to have, and what makes people dream.
Unseen long	Graz is the second largest city in Austria after Vienna. It is located in the southeast of Austria on both sides of the river Mur. The population of Graz is about 300 thousand and it is home to many students. Its historic center is one of the best preserved city centers in Central Europe. Some famous landmarks in Graz include the Castle Hill, clock tower, and town hall.

5.1.1 Fine-tuned model

From each text prompt we generated a mel-spectrogram as well as attention weight matrices for each step of flow. Attention weight plots provide a visual indication of how well the text input has aligned with the mel-spectrogram frames, as explained in section 3.3.

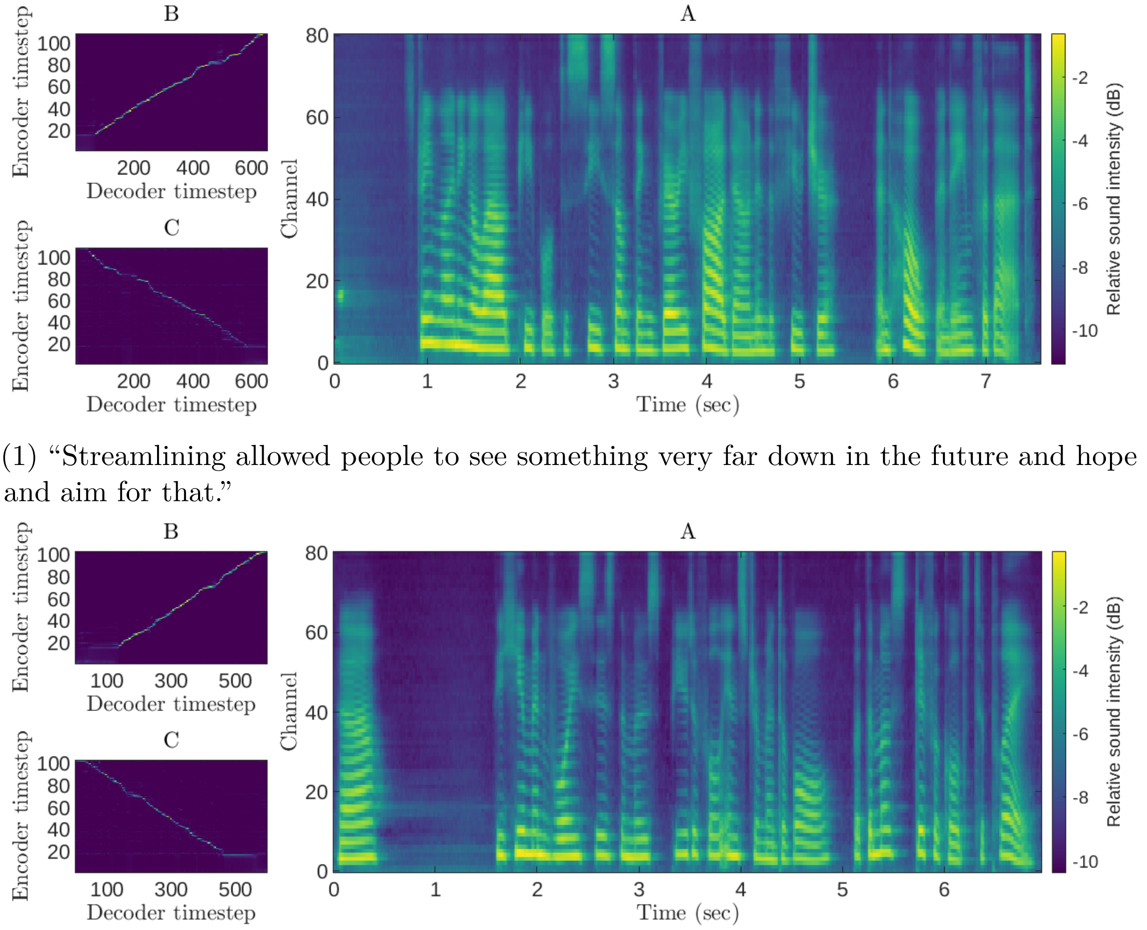
Figure 14 illustrates generated mel-spectrogram plots (panel A) and attention weight matrices (panels B and C) for the seen short (Subfigure 1) and unseen short (Subfigure 2) text inputs. Formant frequency patterns (as explained in section 2.1.2) are clearly visible, especially in the lower frequency areas. The short segments of silence between formant patterns likely represent pauses between words or phonetic units. Attention weight matrices for both text prompts feature a clear diagonal line across the encoder and decoder timesteps, except for some minimal fragmentation in the early timesteps. These results indicate that the model has learned to attend between text and mel-spectrogram relatively well, though some distortion could be present in the early timesteps. There are no major discrepancies between the the mel-spectrograms and attention plots of the two short prompts, which suggests that the model is able to produce output at an equal level for both seen and unseen input.

Figure 15 illustrates generated mel-spectrogram plots (panel A) and attention weight matrices (panels B and C) for the seen long (Subfigure 1) and unseen long (Subfigure 2) text inputs. Attention weight matrices for both steps of flow are almost indistinguishable, which indicates that the model has not learned to align between text and mel-spectrogram frame. We hypothesize that the attention mechanism of the model is not sufficient for decoding very long texts without pauses and produces distorted speech as a result. From the mel-spectrograms alone it is difficult to evaluate whether or not distortion is present because the formant patterns are squeezed into a smaller space due to the longer duration of the texts.

We generated a mel-spectrogram from the actual recording of the target speaker uttering the seen short text prompt to serve as a ground truth comparison. Figure 16 shows the comparison between the ground truth mel-spectrogram (panel A) and its voice clone counterpart (panel B). The ground truth audio is longer in duration and has greater sound intensity even in the higher frequencies. The cloned audio has one second of silence, which perhaps indicates noise or a decoding error. Sections of silence between voiced segments seem to generally be longer in the cloned mel-spectrogram. The formant patterns of ground truth and cloned mel spectrograms seem to follow a similar general pattern, though the details are quite different.

5.1.2 Model trained from scratch

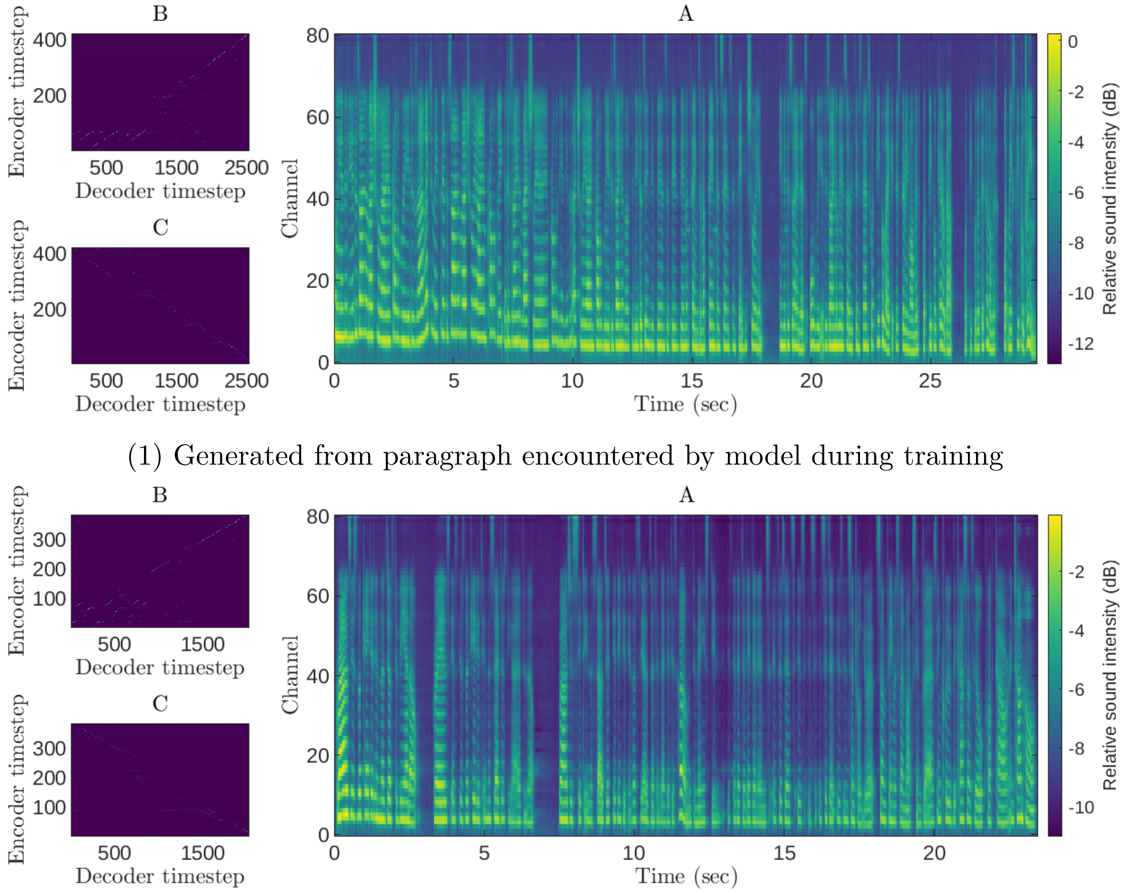
As comparison, we also synthesized mel-spectrograms and attention weights using the model trained from scratch, shown in Figure 17. (1.A) and (2.A) show the generated mel-spectrograms for the seen sbort and unseen short text prompts, respectively,



(1) "Streamlining allowed people to see something very far down in the future and hope and aim for that."

(2) "The human voice is the most beautiful instrument of all, but the most difficult to play."

Figure 14: Synthesized mel-spectrogram (A) and attention weight plots (B), (C) generated from two different sentences: (1) and (2). Sentence (1) was part of the Flowtron fine-tune training process, while (2) was previously unseen by the network. (B) and (C) signify the alignment between text and mel-spectrogram frames for the first and second steps of flow, respectively. Speech formant patterns in both (1.A) and (2.A) are clear and easy to recognize. (B) and (C) in both (1) and (2) indicate good alignment between text input and frame output, although some distortion can be seen in the early timesteps. The network seems to perform at the same level regardless of whether it has encountered the text input before or not.



(1) Generated from paragraph encountered by model during training

(2) Generated from paragraph previously unseen by model

Figure 15: Synthesized mel-spectrogram (A) and attention weight plots (B)(C) generated from two different paragraphs (1) and (2). Paragraph (1) was part of the Flowtron fine-tune training process, while (2) was previously unseen by the network. (B) and (C) in both (1) and (2) appear fragmented and barely visible, which indicates that the model has not learned to align between text and mel-spectrogram frames.

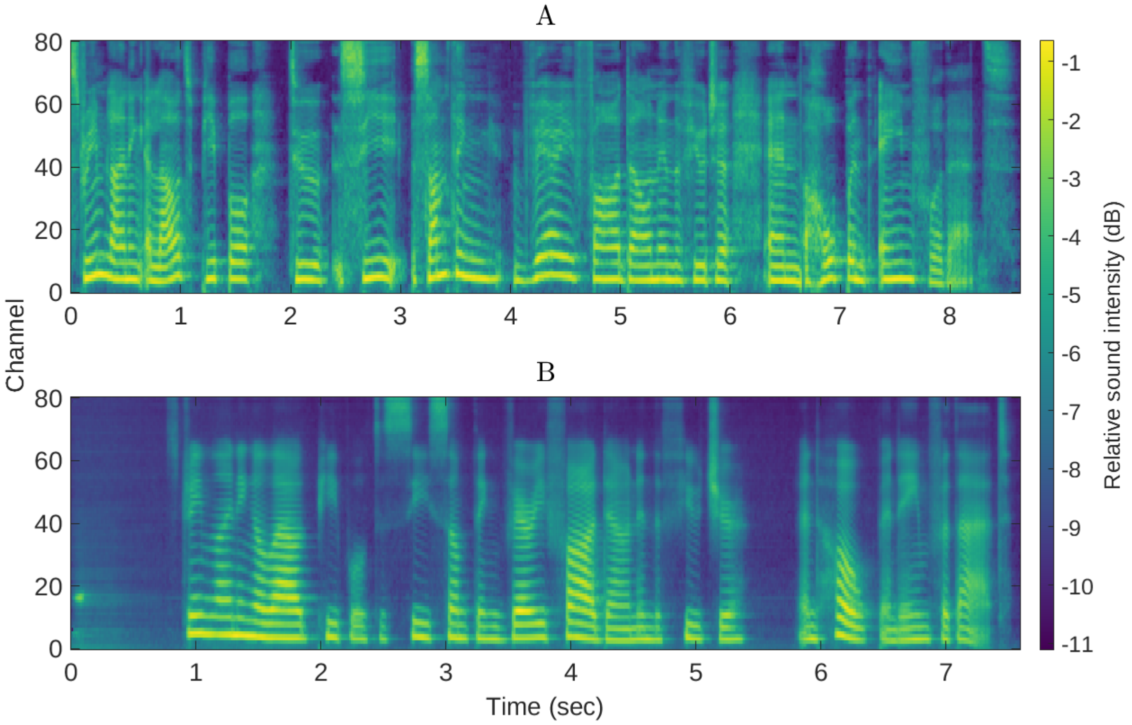
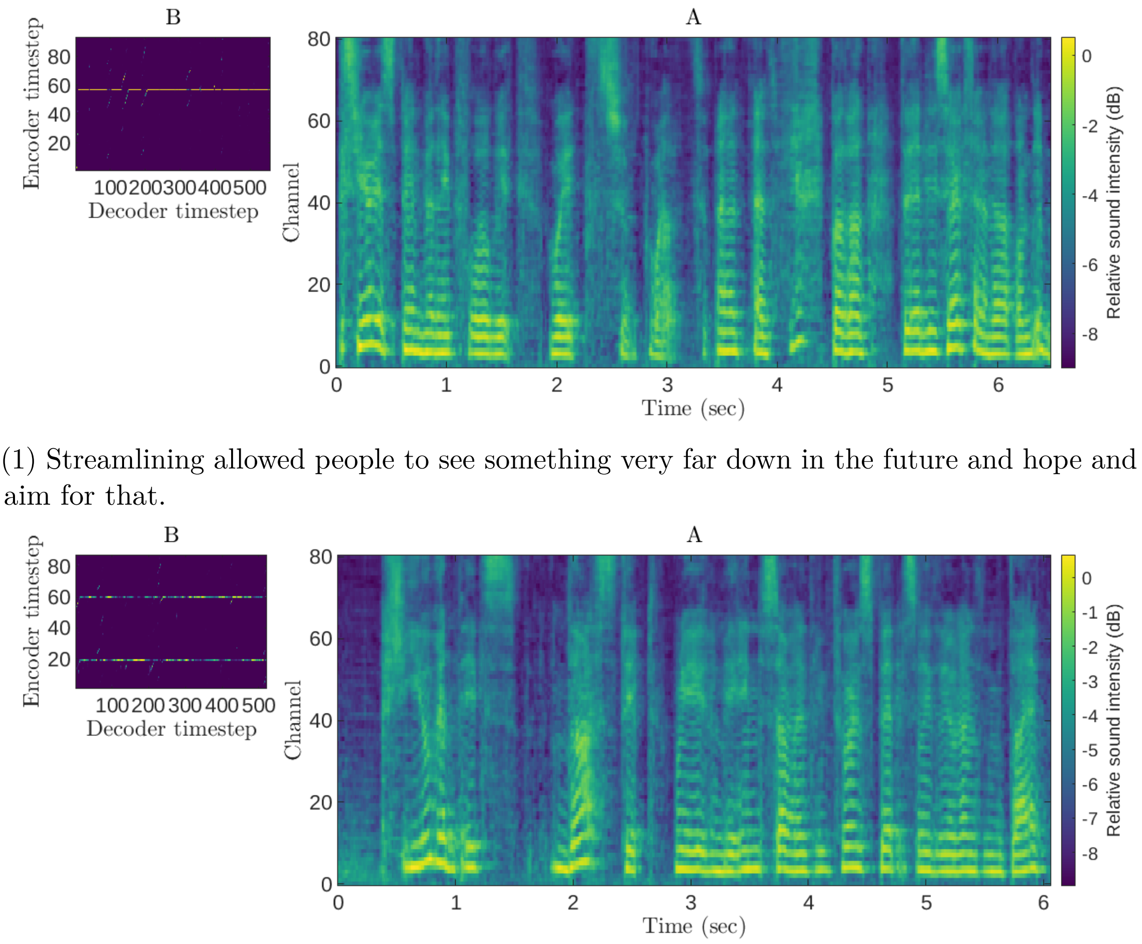


Figure 16: Ground truth (A) and cloned (B) mel-spectrograms for the sentence “Streamlining allowed people to see something very far down in the future and hope and aim for that.“ (A) and (B) differ in duration and relative sound intensity, especially in the higher frequencies. The formants follow the same general pattern.

while (1.B) and (2.B) show the attention weight matrices for the two prompts (the model was only trained for one step of flow; therefore there is only one attention weight matrix per prompt). We were only able to produce results from the short text prompts because the longer text prompts resulted in pure noise as output. In general, the mel-spectrograms seem to be more noisy, as the sound intensity is relatively high throughout the spectrogram in comparison to the spectrogram generated by the fine-tuned model. Interestingly, the model did manage to generate relatively accurate looking formant patterns, but they appear much rougher than those generated by the fine-tuned model. From the attention weight matrices, it is clear that the model has not learned to align between text and spectrogram frame, which was the result we expected from our limited dataset.



(1) Streamlining allowed people to see something very far down in the future and hope and aim for that.

(2) The human voice is the most beautiful instrument of all, but the most difficult to play.

Figure 17: Synthesized mel-spectrogram (A) and attention weight matrices (B), (C) generated from two different sentences: (1) and (2). Sentence (1) was part of the Flowtron training process, while (2) was previously unseen by the network. (1.A) and (2.A) both appear to have a lot of noise, although the formant patterns are still distinguishable. (B) and (C) for both (1) and (2) indicate that the model has not learned to align between text and mel-spectrogram frames due to the limited training dataset.

5.2 Speech synthesis with WaveGlow

We used the pretrained WaveGlow model presented in section 4.1 to decode mel-spectrograms into waveform. The fine-tuned model generates audio output with accurate pronunciation, which verifies the claim in [6] that WaveGlow can be used as a universal vocoder. Speech generated with $\sigma^2 = 0.5$ sounds slightly monotone. In the next sections we explore the effect of varying the expressiveness level. A metallic sound can occasionally be heard in the beginning of each audio, which suggests a decoding error. As hypothesized, the audio samples generated from the longer text prompts sound quite distorted in parts of the utterance. The generated audio repeats words or entire phrases and occasionally sounds like unintelligible mumbles, especially in the beginning and middle part of the utterance. This is likely due to insufficient memory in the attention component of the model. We thus conclude that the most efficient way to use the fine-tuned model is to split a longer text input into sentences before performing inference.

The model trained from scratch with randomly initialized weights seems to adopt the voice characteristics of the target speaker fairly quickly and accurately. As expected, though, the model does not manage to align between text and speech at all. The generated audio sounds distantly like English, but does not contain almost any recognizable words. The algorithm likely does not have access to enough examples of different words and their pronunciations due to the small training dataset and therefore cannot create a general model of the English language. Different speech sounds are simply stringed together randomly instead.

5.3 Speech Variation

Speech expressiveness is controlled by adjusting the variance (σ^2) when sampling z values. When $\sigma^2 = 0$, the model produces outputs fully biased to the model. When σ^2 is increased, the amount of variability in pitch and other voice characteristics also increases, which simulates expressive speech. We calculated the fundamental frequency (F0) contours across the synthesized audio and plotted them in order to visualize the amount of expressiveness. F0 contours are estimated with the YIN algorithm [48] with a minimum F0 of 80 Hz, maximum F0 of 400 Hz, and harmonicity threshold of 0.3. Figure 18 shows a comparison of the F0 contours of the original and synthesized versions of the short seen text. The contours follow a somewhat similar pattern, although the synthesized version is shorter and the cloned audio has less variance in F0. The synthesized speech seems to have a lower F0 throughout the utterance, which can also be heard in the audio as a deeper and more monotone way of speaking.

Figure 19 illustrates the effects of different σ^2 values on the F0 contours of speech synthesize by the same fine-tuned model. As expected, F0 varies more when $\sigma^2 = 1.0$ in comparison with 0.5 or 1.0. The difference between $\sigma^2 = 0.5$ and $\sigma^2 = 0.0$ is not very strong; both contour plots suggest fairly monotone speech.

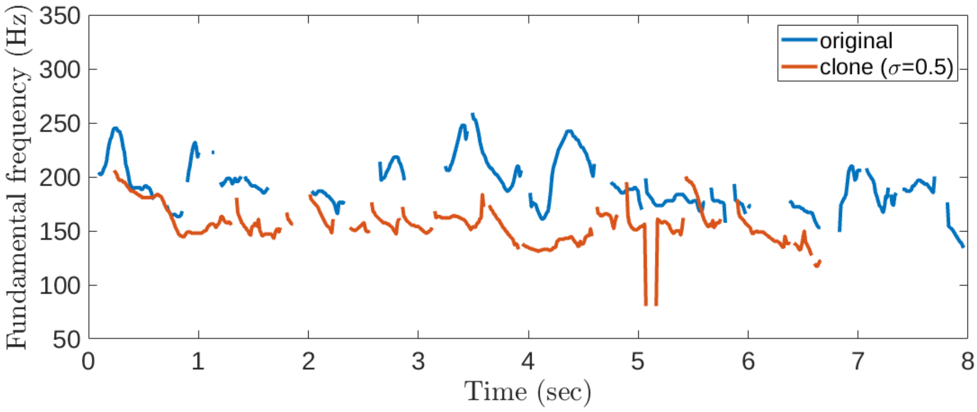


Figure 18: Fundamental frequency (F0) contour patterns of audio generated by a voice clone and its ground truth counterpart (original). The cloned audio has less variance and generally lower pitch than the original.

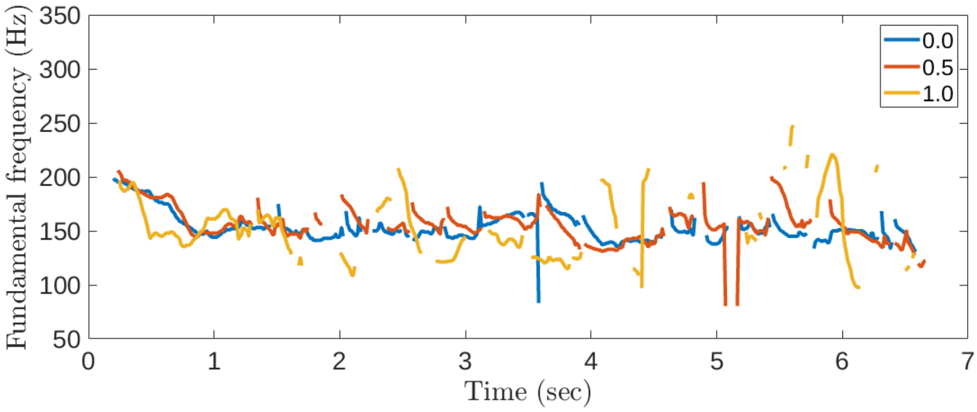


Figure 19: Fundamental frequency (F0) contour plots of cloned audio with varying levels of expressiveness, 0.0 being the least expressive and 1.0 being the most expressive.

Increasing σ^2 also increases variability of speech between different samples of the same text input. One sample may emphasize different parts of the sentence than another sample, and raise or lower the pitch during different parts of the utterance. We generated 10 samples of the same text prompt (seen short) for three levels of expressiveness; $\sigma^2 = 0.0$, $\sigma^2 = 0.5$, and $\sigma^2 = 1.0$ and plotted their F0 contours to observe the variability between samples. The resulting contours are shown in Figure 20. Panel A refers to audio generated with $\sigma^2 = 0.0$. As can be seen in the contours, each of the ten samples follow the same pattern, and are almost indistinguishable from one another. Samples generated with $\sigma^2 = 0.5$ (Panel B of Figure 20) already show more variability between one another, but still follow the same general pattern. The highest variability between samples is seen in clones synthesized with $\sigma^2 = 1.0$

(Panel C of Figure 20). While the contours produced by the two lower levels of expressiveness follow a certain trajectory, the contours produced by the highest expressiveness level look like they could have been generated from different text prompts.

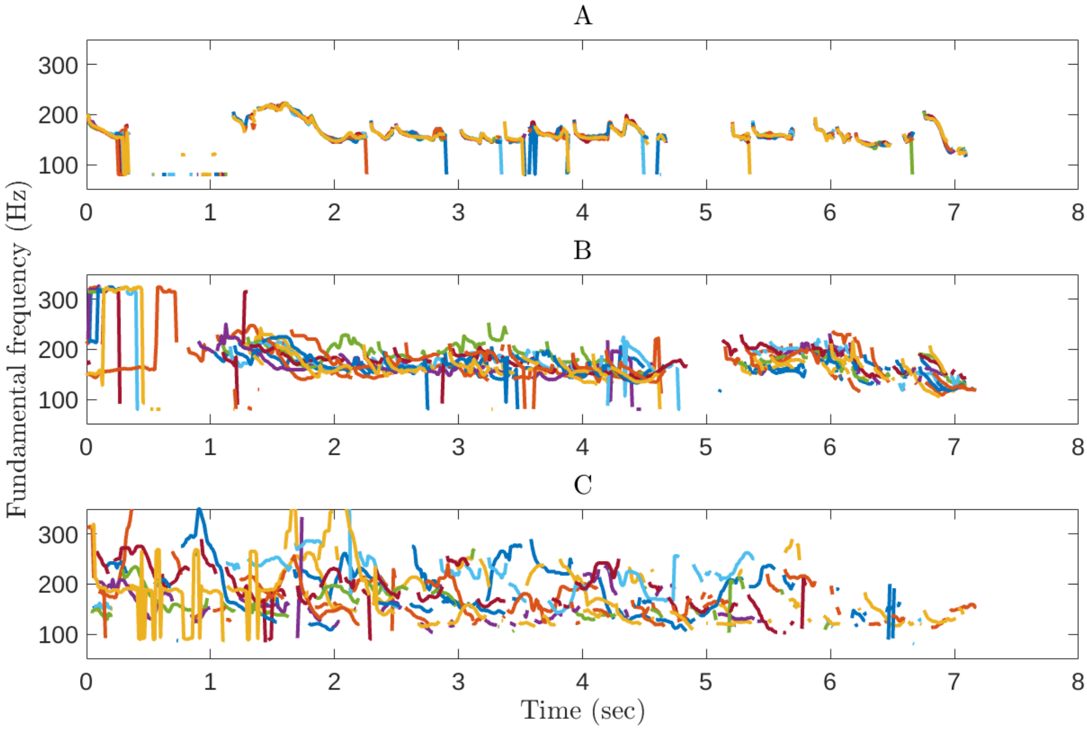


Figure 20: Fundamental frequency (F0) contour plots that show the effect of three different values of variance on F0 variability between speech samples. (A), synthesized with a variance of 0.0, has very minimal variability between samples. (B), synthesized with a variance of 0.5, has slightly more variability, while (C), synthesized with a variance of 1.0 clearly has the most variability between samples.

5.4 Emotion Transfer

In order to produce emotional speech, a posterior distribution conditioned on prior evidence pertaining to each of the four emotion classes is sampled with $\sigma^2 = 0.5$, as explained in section 3.3.2. The sample is then forward passed through the fine-tuned model. We then plotted the F0 contours once again to observe the effect of emotion transfer on F0 contours, shown in Figure 21. F0 varies significantly both throughout the utterance generated in each emotion and between the different emotions. The "angry" emotion has a slightly more monotone contour with a deeper register, which aligns with the expected nature of angry speech. The rest of the emotions, on the other hand, have a relatively high amount of F0 variability within their generated utterances. The fluctuation in pitch could be a characteristic of "amused" speech, for example, but it is difficult to visually evaluate the appropriate amount of F0 variation that embodies a certain emotion.

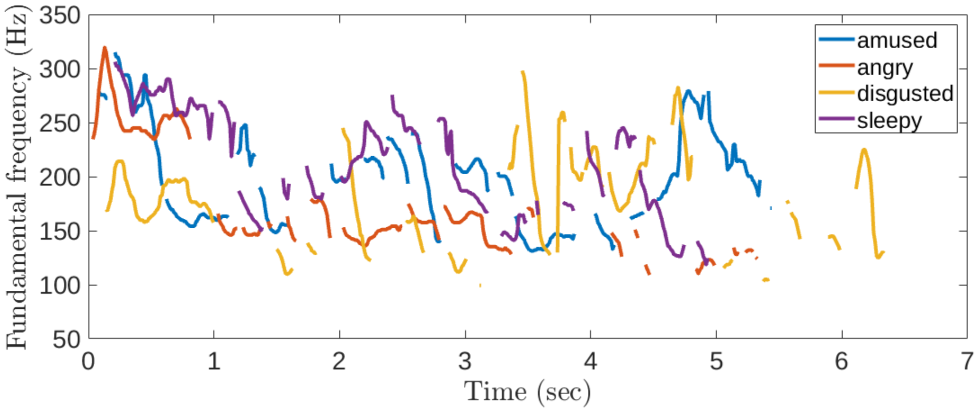


Figure 21: Fundamental frequency contours resulting from voice clone samples conditioned on emotional speech. Emotional transfer introduces a lot of variance in fundamental frequency.

5.5 Mean Opinion Score Tests

A small-scale Mean Opinion Score (MOS) Listening Test was conducted in order to subjectively evaluate the quality of fine-tuned clones. The test had three different parts: (1) listening experience, (2) similarity to ground truth audio, and (3) emotional speech perception. The sample size of the test was 10 people. The requirements included fluency in English and a normal hearing level. Each listener completed the test in a quiet space free of distractions using headphones.

In the first part, listeners evaluate the listening experience of the audio synthesized with the fine-tuned model, generated with five different levels of voice expressiveness: $\sigma^2 = 0.0, 0.2, 0.5, 0.8, 1.0$. Each audio clip was generated with the same text input: “Life isn’t about waiting for the storm to pass, it’s about learning to dance in the rain.” Therefore the only difference between the audio clips is the variance used. Listeners scored their own subjective listening experience using three different indicators: (1) intelligibility, (2) naturalness, and (3) accuracy. Intelligibility refers to how well the listener can understand what is being communicated in the audio. Naturalness is defined as the extent to which the audio resembles human speech. Accuracy asseses whether or not words are pronounced correctly. Each of the indicators were scored on a scale of 1 to 5, where 1 is the lowest perceived experience and 5 the highest perceived experience. Figure 22 summarizes the results of the listening experience section of the listening test. Listeners seemed to give higher scores to speech with higher levels of expressiveness. On the other hand, scores drop slightly with $\sigma^2 = 1.0$, especially in the case of naturalness. While completely monotone speech is definitely unnatural, too much expressiveness and intonation can also be deemed unnatural to the average listener. With the style transfer approach, it is not possible to specify which words or phrases should be expressed in a more lively way. Therefore, when cloned speech is synthesized with too much variability, the parts that are emphasized can seem random, and subsequently, unnatural.

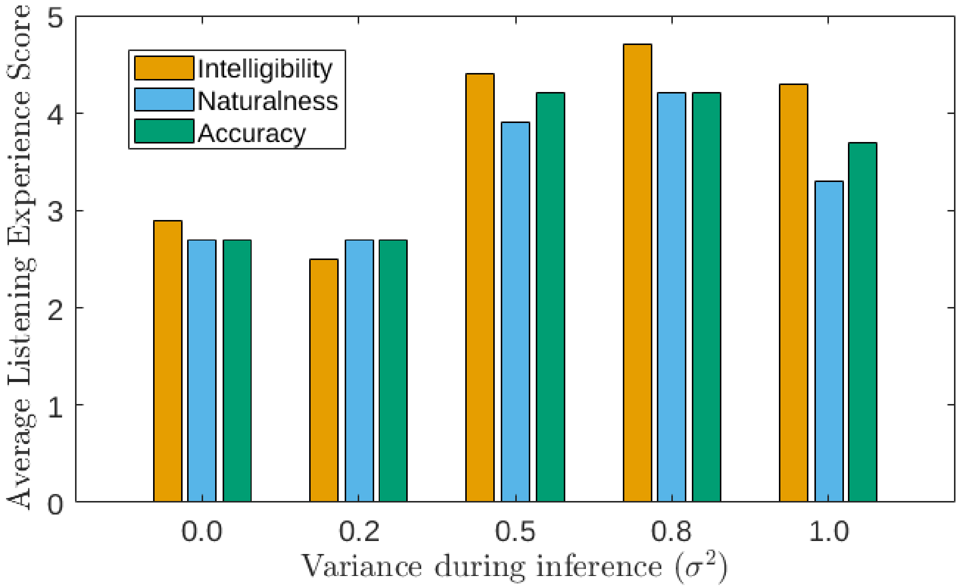


Figure 22: Average listening experience scores of voice clone speech samples synthesized with varying levels of expressiviness. Samples synthesized with variance (σ) values of 0.5 and 0.8 received the highest scores on intelligibility, naturalness, and accuracy.

In the second part of the test, listeners were asked to evaluate the similarity of cloned audio to ground truth audio. The utterance "I grew up in Austria, I'm all Austrian by birth, and by upbringing, and by culture." was synthesized from the fine-tuned model with the same 5 varying levels of speech expressiveness as in the previous section of the test. The listeners evaluated the similarity of the cloned audio to the actual recording of the target speaker speaking the same utterance on a scale of 1 to 10, where 10 indicates that the ground truth audio and the cloned audio are indistinguishable and 1 indicates that the two audios sound like they are spoken by two different speakers. Figure 23 summarizes the results of the second part of the test. Each audio clip received an average score of less than 4, which insinuates that the cloned audio does not bear much resemblance to ground truth audio, regardless of speech expressiveness level.

In the last section of the test, listeners evaluated the perception of emotional speech generated with the style transfer method outlined in section 3.3.2. Listeners were given a total of 8 audio clips, each of which should encompass one of four different emotions: angry, amused, disgusted, or sleepy. The text inputs of the generated audio are "Life isn't about waiting for the storm to pass, it's about learning to dance

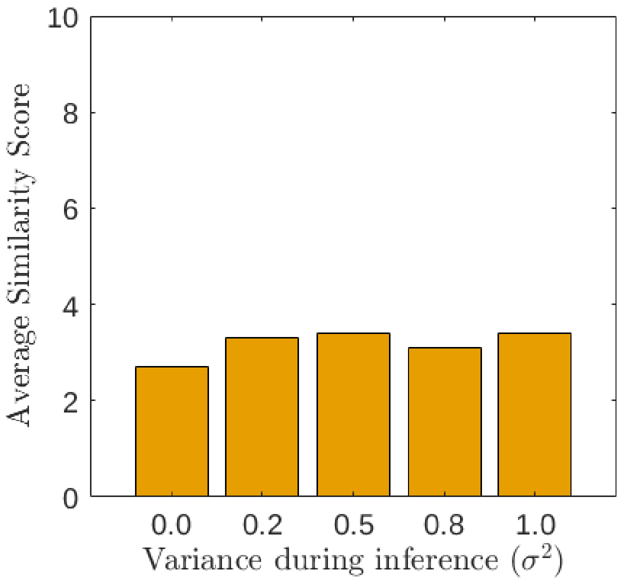


Figure 23: Listeners scored the similarity of cloned audio of different levels of expressiveness to ground truth audio on a scale of 1 to 10. Each sample received an average score of about 3.

in the rain." and "I'm Austrian by birth, and by upbringing, and by culture." The text inputs were chosen to be fairly neutral and thus should not give any indication of the intended emotion through the meaning of the sentence. Listeners were asked to infer the perceived emotion expressed in each audio clip. If none of the listed emotions were recognized in the audio, the listener could also choose "None."

The pie charts in Figure 24 summarize the results of the final listening test section. The expressed emotion was chosen correctly only 5 percent of the time, which indicates that the style transfer method was not successful in conveying emotions through artificial speech. Interestingly, both the "amused" and the "disgusted" voice clone were perceived as "angry" by 40 percent of the listeners. Further, 35 percent of the listeners thought that the "sleepy" clone souded "amused". The low level of agreement with the intended emotion raises the question of how well this method can work and calls for further investigation of both the underlying principle and the used "emotional" voice samples. However, the results also show that performing style transfer results in voice clones that are perceived to express some of the given emotions, even if there is little consensus amongst the listeners on the specific emotion it conveys.

5.6 TTS application

Finally, we developed a simple, user-friendly graphical user interface (GUI) application in order to showcase the results of the trained Flowtron models and to make the

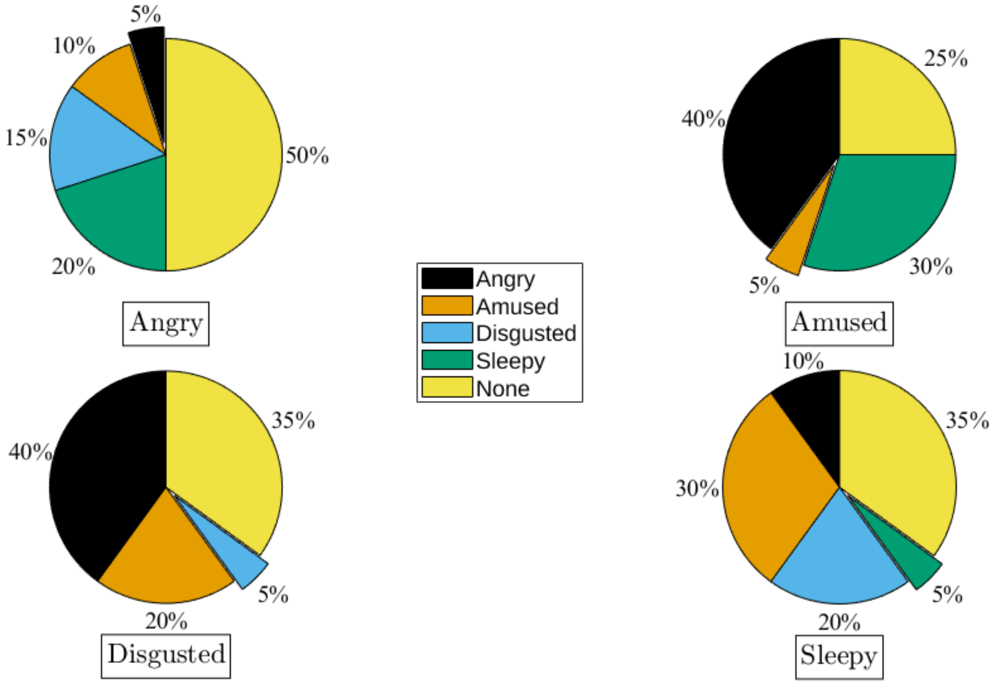
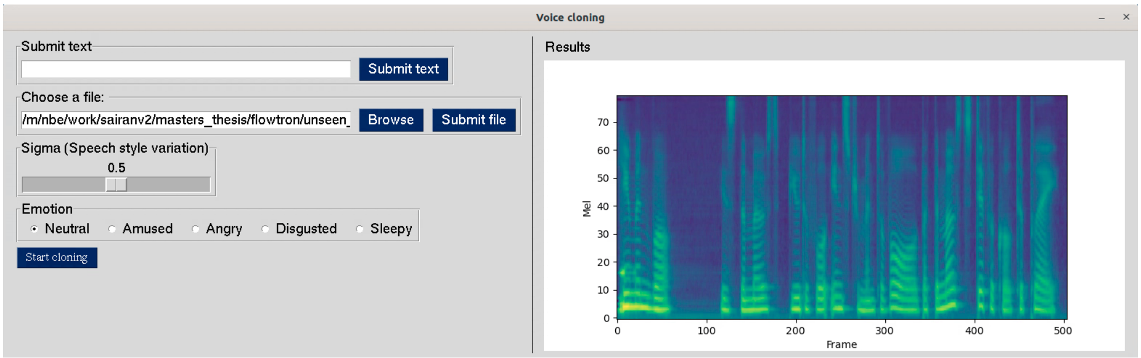


Figure 24: Pie charts which summarize the emotion that listeners guessed was being conveyed in voice clone samples conditioned on emotional speech. The correct emotion was chosen only 5 percent of the time.

technology accessible to a wider audience. The GUI was built with PySimpleGUI[7]. A screenshot of the GUI is shown in Figure 25.


Choose a file: /m/nbe/work/sairanv2/masters_thesis/flowtron/unseen_

Sigma (Speech style variation) 0.5

Emotion: ● Neutral ○ Amused ○ Angry ○ Disgusted ○ Sleepy

Figure 25: GUI application for demonstration purposes: on the left side the user can input text and choose preferred parameters for style of speech. On the right, the generated mel-spectrogram from the input text is displayed automatically.

The user can either directly input text or submit a file containing the desired text to

[7] https://github.com/PySimpleGUI/PySimpleGUI

be synthesized. The user may control the variation of speech by toggling σ^2 to a value between 0.0 and 1.0. The user may also choose a desired emotion they wish to convey in the speech. The default emotion is neutral. The “Start cloning” button locks in all of the chosen input values and parameters and begins the synthesis process. Input text is first split into sentences and inference is performed on each sentence separately to prevent distirted speech. On a CPU, inference takes about 0.77 seconds per character. With a GPU, inference time is roughly half of that. When inference has finished, the program saves both the resulting mel-spectrogram and waveform file into a specified output folder. The audio is automatically played aloud and the reulting mel-spectrogram displayed on the left side of the GUI.

The GUI does not require its users to directly interact with complex code, so it is accessible to a much wider audience. In addition we also aim to give its users a variety of different choices when it comes to the nature of generated speech. Because inference takes a relatively long time, use of the application as a real-time communication device is not practical. Nevertheless, the application serves as an example of what a potential AT tool could look like.

6 Discussion

This thesis demonstrates an application of deep learning based low-resource artificial speech synthesis in the context of people living with vocal impairments. The framework is based on the Flowtron mel-spectrogram synthesizer in combination with the WaveGlow vocoder.

Summary of results. It was found that the fine-tuning method for low-resource voice cloning produce satisfactory speech samples for our data, while models trained on the limited target voice data alone produced unintelligible speech. We concluded that the optimal training setup involves fine-tuning speaker 669 from Flowtron's multispeaker base model trained on 2300 different speakers from the LibriSpeech corpus. We trained the model for 13000 iterations, which is equivalent to about 1500 epochs, after which the validation loss began to plateau.

We conducted a three part MOS listening test to evaluate the voice clone from a subjective point of view. First, we generated five cloned speech samples of five differing levels of speech expressiveness and asked listeners to evaluate the intelligibility, accuracy, and naturalness of each sample. Listeners preferred the fourth highest level of expressiveness, as it scored the highest on all three indicators. For the second part, listeners were asked to score the similarity between ground truth audio and cloned audio samples of the same five levels of speech expressiveness. While scores between samples were varied, each sample received an average score of around 3 points out of 10. For the final part, we conditioned the voice clone model to express four different emotions using the style transfer method outlined in section 3.3.2. We then asked listeners to determine which emotion they believe is conveyed in each sample from a list of options, which included the four emotions as well as an option for none of the listed emotions. For each of the samples, the correct emotion was chosen only five percent of the time, but the answers also varied widely between emotions. We concluded that the style transfer approach does modify the voice clone in a way that would signify emotion, but the type of emotion is not conveyed clearly enough.

Finally, we created a simple computer application which takes text as input and generates audio in the cloned voice as output. The application provides a more intuitive demonstration of the research conducted in this thesis and at the same time serves as a simple example of AT for a vocally impaired person.

Analysis of results. As expected, fine-tuning a base model allowed us to make use of our limited dataset and clone the voice of our target speaker. The ability to control voice expressiveness when synthesizing speech vastly improved the listening experience. We did not reach expected levels of similarity between ground truth audio and cloned audio, likely due to dissimilarities between characteristics of the target voice and the those of the speakers in the base model dataset. While the process of style transfer in the context of emotional speech altered the voice in a noticeable way, listeners were unable to perceive the intended emotion from the speech alone. While the Flowtron and WaveGlow frameworks offer simple and stable training with interesting features, there are some shortcomings when it comes to its use as AT.

Impact statement. The work in this thesis provides a new perspective in the field of artificial speech which highlights the importance of bridging the gap between theory and practice. Not only do we introduce step by step instructions for creating a voice clone from limited personal audio data, but we also present a comprehensive set of guidelines for the evaluation of voice clones for a certain use-case.

Limitations. While artificial speech synthesis with deep learning proved to be a fascinating topic with lots of potential for application in the real world, certain obstacles were met during the pursuit of this thesis. With the vast amount of different artificial speech synthesis architectures available, choosing one that is suitable to this project was not a simple task. The documentation of Flowtron [6] was also confusing at times; it was not always clear whether technical errors and quality limitations originated from the code, the dataset, or from insufficient training time.

Deep learning is very computation-heavy, especially in the case of natural speech networks, which generally require large sets of data and powerful GPUs to perform at the optimum level. Therefore training performance was dictated by data and computation constraints. The training process of Flowtron is simple and stable due to its flow-based approach and likelihood maximizing optimization algorithm. On the other hand, the autoregressive nature of Flowtron leads to slow mel-spectrogram generation, which, in turn, slows down total speech synthesis time by a significant amount. Since long inference time is not convenient for a communication device intended for use in real-time, further research should be conducted to speed up the process. In general, it is important to shorten inference time as much as possible without sacrificing synthesized voice quality.

Since the listening test was limited to a relatively small sample group, the survey results may not be a proper representation of the data. Participants also possibly have different criteria when it comes to scoring audio, but this is inevitable in a questionnaire based on subjective opinions.

7 Conclusion and Outlook

Artificial speech synthesis with the use of deep learning is a very up-and-coming field with new innovations emerging at a rapid pace. Speech synthesis research is not only a fascinating topic, but it also has relevance as a tool for people living with speech impairments. Through the research conducted in this thesis, we hope to raise awareness and focus toward technology that promotes inclusivity. The ability to speak in a way that resembles one's own personal voice can open many doors for people unable to communicate in a conventional manner. The results of this thesis provide a glimpse into the inner workings of deep learning based speech synthesis from the perspective of the average user of the technology.

References

[1] Y. Wang, R. J. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, “Tacotron: towards end-to-end speech synthesis”, in Interspeech, Vol. 2017 (Apr. 6, 2017), pp. 4006–4010.

[2] S. O. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou, “Neural voice cloning with a few samples”, in Advances in neural information processing systems, Vol. 31 (Oct. 12, 2018).

[3] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno, and Y. Wu, “Transfer learning from speaker verification to multispeaker text-to-speech synthesis”, in Advances in neural information processing systems, Vol. 31 (Jan. 2, 2019).

[4] C. Jemine, “Master thesis : real-time voice cloning”, PhD thesis (Université de Liège, June 25, 2019).

[5] Y. Wang, D. Stanton, Y. Zhang, R.-S. Ryan, E. Battenberg, J. Shor, Y. Xiao, Y. Jia, F. Ren, and R. A. Saurous, “Style tokens: unsupervised style modeling, control and transfer in end-to-end speech synthesis”, in Proceedings of the 35th international conference on machine learning (July 3, 2018), pp. 5180–5189.

[6] R. Valle, K. Shih, R. Prenger, and B. Catanzaro, “Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis”, in International conference on learning representations (2021).

[7] C. Grigorovsky, *Locked-in syndrome patients given new revolutionary AI-powered device*, Health Europa, (Aug. 20, 2018) https://www.healtheuropa.com/locked-in-syndrome-new-device/87782/ (visited on 02/27/2023).

[8] P. Norloff, *Locked in syndrome patients helped by eye tracking technology | eyegaze*, (Feb. 15, 2022) https://eyegaze.com/locked-in-syndrome-patients-helped-by-eye-tracking-technology/ (visited on 02/27/2023).

[9] J. A. Chandler, K. I. Van der Loos, S. Boehnke, J. S. Beaudry, D. Z. Buchman, and J. Illes, “Brain computer interfaces and communication disabilities: ethical, legal, and social aspects of decoding speech from the brain”, Frontiers in Human Neuroscience **16** (2022).

[10] R. Prenger, R. Valle, and B. Catanzaro, “WaveGlow: a flow-based generative network for speech synthesis”, in ICASSP 2019 - 2019 IEEE international conference on acoustics, speech and signal processing (Oct. 30, 2018).

[11] P. Roach, *English phonetics and phonology: a practical course*, 4th ed. (Cambridge University Press, Cambridge ; New York, 2009).

[12] P. J. L. illustrator medical medical, *English: head lateral view with mouth anatomy*, Dec. 23, 2006.

[13] P. Ladefoged and I. Maddieson, *The sounds of the world’s languages*, Phonological theory (Blackwell Publishers, Oxford, OX, UK ; Cambridge, Mass., USA, 1996), 425 pp.

[14] T. Bäckström, O. Räsänen, A. Zewoudie, P. P. Zarazaga, L. Koivusalo, S. Das, E. G. Mellado, Mariem Bouafif Mansali, and D. Ramos, *Introduction to speech processing*, 2nd ed. (July 12, 2022).

[15] M. Yavas, *Applied english phonology: yavas/applied english phonology* (Wiley-Blackwell, Oxford, UK, 2020).

[16] P. A. Abhang, B. W. Gawali, and S. C. Mehrotra, "Introduction to emotion, electroencephalography, and speech processing", in *Introduction to EEG- and speech-based emotion recognition*, edited by P. A. Abhang, B. W. Gawali, and S. C. Mehrotra (Academic Press, Jan. 1, 2016), pp. 1–17.

[17] J. Kluk, *Spectrograms of the syllables "dee", "dah", and "doo". the initial formant transitions (formants are highlighted in red) defining the phoneme /d/ demonstrate that apparently different acoustic events result in the same percept.* June 10, 2007.

[18] B. C. Moore, L. K. Tyler, and W. Marslen-Wilson, "Introduction. the perception of speech: from sound to meaning", Philosophical Transactions of the Royal Society B: Biological Sciences **363**, 917–921 (2008).

[19] E. D. Young, "Neural representation of spectral and temporal information in speech", Philosophical Transactions of the Royal Society B: Biological Sciences **363**, 923–945 (2008).

[20] D. C. Peterson, V. Reddy, and R. N. Hamel, "Neuroanatomy, auditory pathway", in *StatPearls* (StatPearls Publishing, Treasure Island (FL), 2022).

[21] C. O'Callaghan, "Auditory perception", in *Stanford encyclopedia of philosophy*, Fall 2021, https://plato.stanford.edu/archives/fall2021/entries/perception-auditory (2021).

[22] G. Pamisetty and K. S. R. Murty, *Prosody-TTS: an end-to-end speech synthesis system with prosody control* (Oct. 6, 2021).

[23] T. H. Tarnóczy, "The speaking machine of wolfgang von kempelen", The Journal of the Acoustical Society of America **21**, 461–461 (1949).

[24] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database", in 1996 IEEE international conference on acoustics, speech, and signal processing conference proceedings, Vol. 1, ISSN: 1520-6149 (May 1996), 373–376 vol. 1.

[25] PaweÅ, Kisielewicz and A. WÅ,odarz, "Speech synthesis systems: disadvantages and limitations", International Journal of Engineering & Technology **7**, 234–239 (2018).

[26] Y. W. Leung, "Computer speech synthesis - a systematic method to extract synthesis parameters for formant synthesizers", PhD thesis (The Chinese University of Hong Kong, 1993).

[27] I. Tokuda, "The source–filter theory of speech", Oxford Research Encyclopedia of Linguistics **29** (2021).

[28] J. O. Smith, *Physical audio signal processing: for virtuell musical instruments and audio effects*, in collab. with Stanford University (Stanford University, CCRMA, Stanford, Calif, 2010), 803 pp.

[29] S. Lemmetty, "Review of speech synthesis technology", PhD thesis (Helsinki University of Technology, Mar. 30, 1999).

[30] D. Klatt, *Invariance and variability in speech processes* (1986).

[31] B. J. Kröger and P. Birkholz, "Articulatory synthesis of speech and singing: state of the art and suggestions for future research", in Multimodal signals: cognitive and algorithmic issues, edited by A. Esposito, A. Hussain, M. Marinaro, and R. Martone (2009), pp. 306–319.

[32] A. Tsukanova, B. Elie, and Y. Laprie, "Articulatory speech synthesis from static context-aware articulatory targets", in ISSP 2017 - 11th international seminar on speech production (Oct. 2017).

[33] S. King, *A beginners' guide to statistical parametric speech synthesis* (The Centre for Speech Technology Research, University of Edinburgh, UK, 2010).

[34] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks", in IEEE international conference on acoustics, speech and signal processing, ISSN: 2379-190X (May 2013), pp. 7962–7966.

[35] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden markov models", Proceedings of the IEEE **101**, 1234–1252 (2013).

[36] L. Hardesty, *Explained: neural networks*, MIT News | Massachusetts Institute of Technology, (2017) `https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414` (visited on 12/27/2022).

[37] A. Kumar, *Deep learning explained simply in layman terms*, Data Analytics, (Sept. 17, 2020) `https://vitalflux.com/deep-learning-neural-network-explained-simply-layman-terms/` (visited on 03/17/2023).

[38] V. Nguyen, *Speech AI concepts you should know*, NVIDIA Technical Blog, (July 26, 2022) `https://developer.nvidia.com/blog/a-guide-to-understanding-essential-speech-ai-terms/` (visited on 11/03/2022).

[39] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: a generative model for raw audio", in ISCA workshop on speech synthesis workshop (SSW 9) (Sept. 19, 2016).

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need", in Advances in neural information processing systems, Vol. 30 (2017).

[41] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, R. A. Saurous, Y. Agiomvrgiannakis, and Y. Wu, "Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions", in 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP), ISSN: 2379-190X (Apr. 2018), pp. 4779–4783.

[42] G. Papamakarios, T. Pavlakou, and I. Murray, "Masked autoregressive flow for density estimation", in Advances in neural information processing systems (June 14, 2018), pp. 2338–2347.

[43] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP", in International conference on learning representations (Feb. 27, 2017).

[44] D. P. Kingma and P. Dhariwal, "Glow: generative flow with invertible 1x1 convolutions", in Advances in neural information processing systems, Vol. 31 (July 10, 2018).

[45] O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language", in Advances in neural information processing systems, Vol. 28 (2015).

[46] A. Adigwe, N. Tits, K. E. Haddad, S. Ostadabbas, and T. Dutoit, *The emotional voices database: towards controlling the emotion dimension in voice generation systems*, arXiv:1806.09514 (June 25, 2018).

[47] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books", in 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP), ISSN: 2379-190X (Apr. 2015), pp. 5206–5210.

[48] A. de Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music", The Journal of the Acoustical Society of America **111**, 1917–1930 (2002).