Data Efficiency in Neural Stylistic Speech Synthesis

Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science in Computational Linguistics
by Research

by

Nishant Prateek
201225113
nishant.prateek@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
NOVEMBER 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Data Efficiency in Neural Stylistic Speech Synthesis" by *Nishant Prateek*, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Advisor: Prof. Manish Shrivastava

To my parents, who never lost hope that I would eventually finish writing my thesis.

Acknowledgments

I would like to thank my parents, Mrs. Seema Ranjan and Mr. Prabhat Ranjan for their unwavering love, support, and patience while I embarked on my research journey. They have been my constant cheerleaders, and have kept faith in my potential even when I lost mine. They have always been there to lend a listening ear, offer a helping hand, and provide a shoulder to lean on during difficult times.

I would also like to thank my advisor, Dr. Manish Shrivastava for the insightful discussion, endless patience while I toyed with various unsuccessful and successful research ideas, and occasional stern reminders to write this thesis. From him I learnt the art of turning ideas into concrete research problems. Some of our early discussions have now formed the core of the research focus in my professional career.

I would like to extend my gratitude to Dr. Amy Ogan, and Dr. Kishore Prahallad, my early guides into the world of research as a measly second-year undergraduate student. They instilled in me the intellectual curiosity, critical thinking, and skills in systematic and rigorous investigation required for academic research.

I was fortunate to start my professional career in research-focused teams as Amazon and Citi Innovation Labs. My colleagues here have shared their expertise, been a fountain of ideas, and a source of motivation. To all my present and ex colleagues, thank you.

Last but not the least, I'd like to thank my friends at IIIT-H and the colleagues at LTRC for wonderful discussion, countless memories, and occasional distractions when needed.

Abstract

Recent developments in neural text-to-speech synthesis (NTTS) have been able to produce high-quality human-like speech samples. However, these models require large amounts of studio-quality recordings for training. As a result, training NTTS models to serve multiple speakers and styles is resource intensive, both in terms of time and compute costs.

In this thesis, we discuss the data-efficiency methods for training an NTTS acoustic model with limited data. We explore several data-efficiency techniques for speaker and style modelling with limited data. We discuss multi-speaker training for NTTS model with limited data for each speaker, and speaker adaptation for fine-tuning a pre-trained multi-speaker NTTS model with few minutes of training data for the target speaker. Controllability is TTS systems refers to the ability of explicitly controlling the prosodic variations of the synthesised speech. We explore controllability in NTTS with latent-variable conditioning with variational autoencoders (VAE). These can be used for one-shot style transfer from a reference speech sample. We further discuss improving the posterior flexibility of latent-variable from VAE using normalising flows.

We adapt the multi-speaker training strategy to generate newscaster style speech with limited stylistic training data. We first analyse prosodic variations in the neutral style, newscaster style, and mixed expressive corpora. From this, we conclude that the newscaster style is more dynamic than the neutral style, however with lower dynamic range and prosodic variations than a mixed expressive corpora containing recordings with different emotions. The problem of generating newscaster style of speech with limited training data is posed as that of creating a bi-style model in which a one-hot style ID can be modified to generate either neutral or newscaster style speech. We only use a quarter of the data for newscaster style as opposed to that of neutral style. Combining the two styles gives the model enough volume of training data to learn the textual and acoustic alignments, while only a fractional amount of stylistic data is required to factorise the two styles.

To further improve on the naturalness, we condition the NTTS acoustic model with contextualised word embeddings (CWE). This gives the model additional syntactic and semantic context on the input text. The proposed bi-style NTTS model conditioned is shown to improve on naturalness and style-appropriateness for newscaster speech over both neutral NTTS and concatenative systems in MUSHRA evaluations conducted with expert listeners.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Speech is the most widely-used and the most reliable form of human communication. The ability to communicate through speech is naturally acquired, doesn't require specialised tools for production and comprehension in most humans, is omni-directional, and is culturally ubiquitous. This makes speech a prime candidate as an interface for accessible human-machine interaction. Modern AI assistants like *Alexa*, *Cortana*, *Siri*, *Google Assistant*, *Bixby*, and *Clova* use speech as the primary mode of interaction. In addition to lowering the barrier of usability for AI technology, speech interfaces are powerful accessibility tools. Speech interfaces have wide-range of applications - voice search, translation, communication with smart devices, meeting summarisation, medical transcription, Interactive Voice Response Systems (IVRS), travel announcements, audiobooks, reading aloud websites and labels for the visually impaired, brain-computer interfaces, and advertising.

Speech, like other forms of communication, is symmetric, i.e. it requires perception of communicated information, and transmission of information through intelligible generation. Automatic speech recognition (ASR) systems convert speech from audio waveforms to lower-level representations, typically text. Speech synthesis systems generate speech waveforms given a lower-level representation. Text-to-speech (TTS) is a sub-field of speech synthesis, which focuses on generating intelligible and natural-sounding speech from text inputs.

1.1 Speech beyond linguistics

1.1.1 Information conveyed through speech

Communication is the act of transferring information from one entity (the *sender*) to the other (the *receiver*). Communication has informative elements and communicative elements [62].

Informative element of communication is the information that the sender conveys, regardless of the sender's intention.

Communicative element is the information that the sender intentionally conveys to the receiver.

In the context of speech, the sender of the information is known as the *speaker*, and the receiver knows as the *listener*. Speech conveys a lot more information than the linguistic content it encodes. Spoken language consists of *linguistic*, *paralinguistic*, and *extralinguistic* information [23, 57, 98].

- **Linguistic information:** Information that the speaker intends to convey in explicit verbal form. Linguistic information uses both the phonetic and syntactic rules of language. It is the primary communicative element of speech.

- **Extralinguistic information:** Residual information after the communicative elements of speech are removed. Extralinguistic information encodes the long-term and habitual characteristics of the speaker e.g. gender, age, identity, pitch (high or low), accent, and voice quality.

- **Paralinguistic information:** Information that conveys the speaker's current affective or emotional state, attitude, social setting etc. Paralinguistic information encodes non-verbal communicative element of speech.
  The paralinguistic information in speech is also called its ***prosody***. In speech signal processing, prosody is characterised by the variation pitch or fundamental frequency (*f0*), intensity (measured by amplitude), duration, and rhythm.

It is important for TTS systems to encode linguistic, paralinguistic, and consistent extralinguistic information for it to better engage its users. Researchers in speech synthesis evaluate their systems on *intelligibility* and *naturalness*. If the synthesised speech conveys linguistic content without any loss of information, it is deemed intelligible. For a speech output to be deemed natural, it will need to convey the desired paralinguistic and consistent extralinguistic information.

1.1.2 Speaking style

Speaking style defined informally is the "way of speaking" in a linguistic environment or a social context [4]. Despite the actual verbal content, the listener may still be able to distinguish among a book being read aloud, a newscast, a formal interview, an informal conversation between friends, or a monologue. Essentially speaking style forms the paralinguistic information conveyed through speech.

Eskenazi [25] analysed several speaking styles, and have classified speaking styles based on speech variations in four categories:

- Voice qualities - *breathy, creaky, whispery modal, and tense.*

- Speaking rate - *fast, very fast, and slow.*

- Dimensions of speaking styles - *careful, clear, formal, conversational, spontaneous, connected, scripted, unscripted, normal (neutral), reading, and laboratory.*

- Specific tasks - *newscast, sports, professional, and interview.*

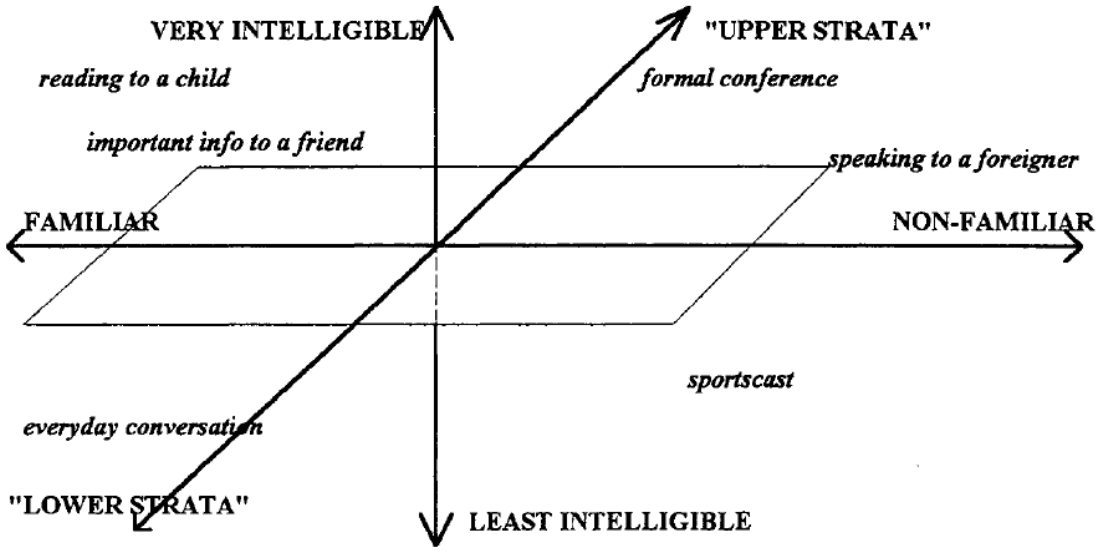

Figure 1.1 Speaking style hyperspace. From [25]

Based on the analysis above, Eskenazi also suggested that most speaking styles can be represented as points in a hyperspace with three axes representing intelligibility, social strata, and familiarity [25].

In figure 1.1 Eskenazi have attempted to position some common speaking styles into the style hyperspace. However, this analysis is limited in terms of speaking styles, and the number of speakers and variations for each style considered. It is still unclear whether all speaking styles can be positioned as points in a speaking style hyperspace, or styles are arbitrary with no formal relation that can defined between different speaking styles.

Several studies have attempted at more formal characterisation of speaking styles using segmental (voicing/devoicing, phonological changes like - phoneme insertion/deletion, schwa deletion, coarticulation etc.) and suprasegmental (amplitude, pitch, duration, speaking rate etc.) features of speech [24, 60]. For example, careful speech has larger duration than casual speech, but is faster than read speech, and read speech has more dynamic pitch range than careful speech. Speakers also tend to have more phonological changes in casual speech than read speech [24].

1.2 Text-to-Speech

Text-to-speech systems model the conditional probability of acoustic features given an input text in the form of a phoneme or character sequence. Conventional TTS systems consist of a front-end component which generates the linguistic features from a given sequence of text, and a back-end component generates audio waveforms from the given linguistic features. We describe the front-end and back-end components of traditional TTS sytems below:

1.2.1 Front-end component

The front-end component of TTS systems consists of a pipeline of text-processing components and annotates text with specific linguistic information that can be processed by the back-end component for generating speech.

First, the text is normalised - text is converted to lowercase, tokenized, numeric and alpha-numeric components are converted to text, and the acronyms and abbreviations are expanded. Some front-end components also involve part-of-speech tagging to resolve word ambiguity.

The words are then converted into their phonetic transcriptions using a pronunciation dictionary (such as the CMU Pronunciation Dictionary [1]). The words that are not present in the dictionary are converted into phonemes using grapheme-to-phoneme (G2P) or letter-to-sound (LTS) systems.

Phonemes are the smallest units of sound that distinguish one word from the other. The CMU Pronunication Dictionary uses a set of 39 phonemes, called the phone-set. For more natural sounding speech, the processed text is annotated with prosodic markers. This can include information about the pitch contour, amplitude, stress, segment duration, pauses, and voice quality.

1.2.2 Back-end component

The back-end component of a TTS system is responsible for generating speech waveform given the linguistic information provided by the front-end system. There are broadly three paradigms of back-end systems:

- **Concatenative or unit-selection systems :** Concatenative or unit-selection system [35] generate speech by selecting and joining smaller units of speech from a large speech database. In most systems, these units are diphones. Diphones are combination of speech waveforms of two phonemes, starting from the middle of one phoneme and ending at the middle of second phoneme. The selection is done such that the concatenated units are similar to the surrounding units, and there's little or no perception discontinuities at the point of concatenation. This is done by minimizing a combination of the target cost and concatenation cost functions.

  Concatenative systems produce natural sounding speech because the units are actual recordings of human voice. However, they suffer from the following limitations:

  - Data requirement: The quality of the generated speech is dependent on the size of the recorded speech database. A large database is required to ensure converage of all possible diphones.
  - Flexibility and controllability: Concatenative systems also have difficulty in handling prosodic variations, as the units in the pre-recorded database might not contain the variations in the intended prosody. Furthermore, it is difficult to manipulate the prosody of the pre-recorded units.

[1] Can be accessed at http://www.speech.cs.cmu.edu/cgi-bin/cmudict

- Scalability: Concatenative systems are also not scalable as these do not work on multi-speaker, multi-style, or multi-lingual settings.

- **Statistical Parametric Speech Synthesis (SPSS) :** SPSS systems [11, 134] are inverse of ASR systems, such that these generate series of acoustic representations, frame-by-frame (each frame is a speech segment of a fixed duration, e.g. $10ms$), given text or linguistic features as inputs. These acoustic representations are then passed on to a waveform generation module, called *vocoder*[83, 69, 42], that outputs speech waveforms. SPSS are more flexible than concatenative or unit-selection systems, because they require lesser training data, and the speech characteristics of the output are not just limited to that of the recorded speaker in the database.
  SPSS systems provide more flexibility and controllability than concatenative TTS systems. However, they rely on hand-crafted acoustic and linguistic features, and the output produced is not as natural as that of concatenative TTS.
  Hybrid unit-selection (HUS) systems [127, 66] combine both SPPS and concatenative systems by using parametrical models (like Hidden Markov Models or neural networks) to predict acoustic properties that inform selection of speech units. However, the problem of scalability persists.

- **Neural Text-to-speech (NTTS) :** NTTS systems use neural networks, like sequence-to-sequence models [119, 102], or transformers [59, 58] to convert text into lower-level acoustic representations. These representations are then converted into speech waveforms using vocoders [39, 61]. Neural text-to-speech systems are more flexible than concatenative and SPSS systems, can model complex segmental and supra-segmental variations in speech, and produce more natural sounding outputs. However, NTTS systems usually require significantly more training data [56], are expensive to train, and require longer processing times to generate speech waveforms.

1.3 Need for Data-efficient Neural Text-to-speech

Neural text-to-speech (NTTS) models are capable of generating high-quality speech almost indistinguishable in quality from human speech. However, these models are data-hungry and require tens of hours clean studio-quality recordings, with extensive phoneme-coverage, to train a speaker, style, or language-dependant TTS models. [56]. Gathering training data in such magnitude is both expensive and time-consuming, and in some case impossible.

With the advancement of speech interfaces, there's a growing need for TTS models to support multiple contexts - voices, speaking styles, emotions, and languages. The choice of these context enable better user experience, and deliver more engaging conversation with context-appropriate synthetic speech.

As of 2023, there are over 7100 identified spoken languages in the world [22]. Most of these languages historically haven't enjoyed the same attention from speech researchers as the 23 most popular languages which account for more than half of the world population. Additionally, 42% of these are endangered with fewer than 1000 living speakers. The challenge of collecting tens of hours of train-

ing data for these languages is an obstacle in providing access to information to the speakers of these languages, specially the ones with inability to read and write.

Our focus in this thesis is specifically speaker and style variations in NTTS systems. We discuss transfer learning [56, 87], meta-learning [70], and voice-cloning [2] for generating synthetic speech for speakers and styles with limited data. Although beyond the scope of this thesis, these techniques can also be adapted for multilingual NTTS [136, 75, 13]. This helps in scaling NTTS systems to low-resource languages.

1.4 Contributions

NTTS systems require vast amounts of clean studio-quality recordings as training data for generating high-quality speech waveforms. Procuring the training data for each speaker and style context is resource-intensive. With this context, the core contributions of this thesis are as follows:

- We give an overview of data-efficiency methods for speech syntheis. Specifically, we look at training NTTS models in a multi-speaker and multi-style settings. We also look at Bayesian Optimisation techniques for few-shot speaker adaptation, starting from a pre-trained multi-speaker model. We also explore latent-variable conditioning using variational autoencoders (VAE) for one-shot style-transfer, and improving posterior flexibility in VAE using normalising flows.

- We propose a bi-style NTTS model for synthesising newscaster style utterances with one-fourth the data required for training a style-dependent NTTS model. The proposed model can generate both neutral and newscaster style utterances. We show that our neutral utterances generated by proposed model outperform a neutral concatenative TTS system on both prosody and segmental quality. We also show that the proposed model can generate newscaster style utterance with dynamic prosodic variations, and the two styles can be factorised with just a one-hot style ID.

- To improve the naturalness of the synthesised newscaster style utterances, we propose conditioning the bi-style NTTS model with contextualised word embeddings (CWE). For this we introduce an additional CWE encoder as a conditioning network in the NTTS acoustic model. The proposed bi-style NTTS model receives multi-scale conditioning on the input text, with phoneme-level conditioning through the phoneme encoder and word-level conditioning from the CWE encoder. We show that with CWE conditioning we can improve the naturalness and prosody modelling on the bi-style NTTS model without CWE conditioning.

1.5 Thesis Organisation

This chapter introduces the information conveyed through speech beyond the linguistic content, and defines the terms *style* and *prosody* which form the overall theme of this thesis. It also formalises the

problem of text-to-speech (TTS) and gives an overview of different TTS paradigms. The rest of this thesis is organised as follows:

Chapter 2 introduced Neural Text-to-speech (NTTS) and gives a detailed overview of the components in NTTS. We discuss each step in the NTTS pipeline required to generate raw audio waveform perceptible by humans from unprocessed text inputs. This chapter also introduces the evaluation strategies and metrics used for NTTS models.

Chapter 3 focuses on the acoustic model of NTTS system and discusses the large-scale training data requirements for training NTTS systems. We also look at common techniques used for data-efficiency in training NTTS models. We discuss multi-speaker training of NTTS model by combining limited training data from several speakers to build a robust acoustic model. These models however need to be trained from scratch each time a new speaker is introduced. We also introduce Bayesian Optimisation (BO) techniques for speaker adaptation for fine-tuning a pre-existing multi-speaker NTTS model, with just few minutes of training data for the new speaker. The multi-speaker models generate an averaged prosody for each speaker in its training set. This produces unsatisfactory results for stylistic speech synthesis, specially in long-form content. To solve this, we discuss latent variable conditioning from a reference speech signal into an NTTS model for controlling the prosodic variations in synthesised speech. We also discuss improving the posterior flexibility of the latent variable model for one-shot style transfer from a reference speech sample.

Chapter 4 adapts the multi-speaker setting for training NTTS acoustic model to generate newscaster style utterances with limited data. For this, we propose a bi-style NTTS model trained with combined neutral and newscaster style utterances from the same speaker. The proposed model synthesised both neutral and newscaster style utterances with improved naturalness over the concatenative TTS system. The experiments in this chapter also show that the styles can be factorised by using a one-hot style conditioning, without the need for a reference speech sample from the target styles.

Chapter 5 discusses the relation between syntax and semantics of the text and the prosody of the verbalised utterance. In this regard, we proposed conditioning the bi-style NTTS acoustic model with contextualised word embeddings (CWE). We use CWE generated from unsupervised pre-training of language model on a large-scale dataset to condition the NTTS model. The CWE conditioning gives the NTTS model additional word-level and sentence-level context, thus improving the performance on prosody modelling. We also present detailed objective and subjective evaluation to measure the performance of the bi-style NTTS model with and without CWE conditioning.

Finally, **Chapter 6** concludes thesis with the summary of the results and discussions in this thesis, and also discusses the limitations of the proposed model and scope for future work.

Chapter 2

Neural Text-to-speech

Neural Text-to-speech systems replace several components of conventional TTS systems with neural networks. **Neural SPSS** systems [130, 122] have multi-stage modelling process for waveform generation. They use an external front-end model to predict linguistic features. These linguistic features are input to a duration model that predicts the phone-duration for each phoneme. The duration models use recurrent neural networks to predict the phone-duations [133, 131]. The linguistic features are then up-sampled to frame-level and input to the acoustic model which uses Long-short Term Memory (LSTM) networks [32] to predict vocoder parameters for each frame [130]. A signal-processing vocoder is then used to generate waveforms.

Wavenet [78] uses a fully auto-regressive model to predict waveform taking linguistic features, phoneme-durations and predicted acoustic parameter (like *f0*) from an existing vocoder as input, and generates natural-sounding waveforms. It is essentially a combination of the acoustic model and the vocoder. Wavenet has dependence on external hand-crafted linguistic features, and acoustic features for its inputs. Wavenet also is slow and computationally expensive due to its auto-regressive nature, and is impractical for real-time speech synthesis. Parallel Wavenet [77] uses Inverse Auto-regressive Flows (IAFs) [46] and probably density distillation [31] process to address the speed of waveform generation with Wavenet. However, the problem of dependence on external features still remains.

Sequence-to-sequence NTTS models are an attempt towards end-to-end models for TTS. For the rest of this thesis, we will refer sequence-to-sequence models as NTTS. NTTS models are capable of taking words, or characters as inputs to predict vocoder parameters, typically mel-spectrograms. Even though NTTS models still require an external vocoder to generate waveform, they reduce the dependence on hand-crafted features for speech synthesis, by having implicit linguistic feature extraction, duration modelling, and acoustic modelling. Char2wav [104], Tacotron [119], and Tacotron 2 [102] models use normalised grapheme (raw text and characters) inputs. These systems have an encoder module that encodes the character inputs into rich intermediate representations. The attention module aligns the intermediate text representations to acoustic features (mel-spectograms) during training and inference. The alignment process enables us to work with different sequence lengths between the text inputs and the acoustic features. The decoder uses the text-representations and alignment information

to produce mel-spectrograms. A neural vocoder [61] is used to convert the output acoustic features to speech waveform. The next sections will cover the individual components of NTTS in detail.

2.1 Preprocessing

The training data for NTTS models are $\langle text, audio \rangle$ pairs. Each pair is a text sequence, typically a sentence or a paragraph, and its corresponding recorded audio waveform. Before these can be used to train an NTTS model, the $\langle text, audio \rangle$ pairs need to be preprocessed.

2.1.1 Linguistic preprocessing

NTTS systems are capable of taking raw-text as inputs. However, the text needs to be normalised. The first step is tokenisation. Tokenisation splits the longer text sequence into individual words or word-like units (e.g. numbers, dates, currency symbols, abbreviations etc. are separated into individual tokens.

This is followed by classification of tokens into categories. The categories may include standard words, dates, punctuation, emojis, currency, time, distances etc. Based on the classified category the normalisation process includes different rules for the verbalisation of the non-standard words. Recently, neural sequence-to-sequence models [128, 37, 135] have been proposed for text normalisation. These models treat normalisation as machine-translation process. The normalised text is then broken down into characters before finally being input to the model.

Even though NTTS models can directly use character or grapheme inputs, there are certain limitations that come along with such inputs. It has been shown [108] that even though grapheme inputs are able to model pronunciation implicitly, the realisation of phonemes is unreliable and dependent on the size of data for contextual modelling of phone-realisation. Using phonemes as input gives us the ability to control pronunciations, while also significantly reducing the amount of training data required.

In this thesis, we will be using phonemes as linguistic inputs. After normalisation, we use a proprietary G2P system to convert text into phonemes, stress markers, and punctuations that are encoded into one-hot vectors.

2.1.2 Audio Preprocessing

Ideally end-to-end TTS synthesis should be able to directly produce raw audio waveforms for the syntheised speech. Raw audio waveforms are continuous time-domain representations of speech. To convert these continuous waveforms to a discrete representation, the waveforms need to be sampled. Each sample is a reading of the amplitude of the waveform at a certain time-step.

Sampling frequency refers to the average number of samples taken in one second. According to the Nyquist-Shannon theorem [101] the sampling frequency should be at least double the highest frequency in the audio waveform to accurately capture the information. Human ear can perceive frequencies

between $2kHz$ and $20kHz$. To accurately capture all the information in an audio waveform, $40,000$ samples per second need to be recorded. Most TTS systems work with sampling frequency upto $24kHz$ as it has been shown to provide a good trade-off between quality and compute performance [1]. Even with a $24kHz$ sampling frequency, the NTTS systems need to produce $24,000$ samples per second. This requires enormous memory and makes training and inference computationally expensive.

Mel-spectrograms

Frequency-domain representation gives us the ability for compact representation of speech. Fast-Fourier Transform algorithm [76] converts raw-audio from time-domain into frequency-domain. The frequency-domain representation, also called the **spectrum** is graph between the individual frequencies contained in the audio and their corresponding amplitudes.

The raw-audio can be split into shorter overlapping time-segments, called **frames**. For each frame the spectrum is computed and the resulting spectrums is stacked together. This process is called the **Short-Term Fourier Transform (STFT)**. STFT return both the magnitude and the phase information of the speech signal. The phase information is discarded, and the result of STFT is a 3-dimensional representation of audio containing the frequency, amplitude, and time. The frequencies are quantised into equally-spaced bands for easier representations. The resulting output is the **spectrogram** of the audio.

The human-perception of frequencies is not linear. Humans tend to distinguish between lower frequencies better than the higher frequencies. Mel-scale [107] is a logarithmic-function that attempts to replicate the human perception of sound. The frequency, f, is converted into corresponding mel, m, using the following equation [80]:

$$m = 2595.\log_{10}\left(1 + \frac{f}{700}\right) \tag{2.1}$$

The mel-frequencies are then quantised into bands by applying overlapping triangular filterbanks. The filterbanks are designed such that their centers are equally spaced on the mel-scale, but logarithmically scaled on the frequency-scale. It is common-practice to choose 80 or 128 mel-bands for speech synthesis applications.

The mel-spectrogram can be visualised as an image such that the horizontal axis represents time, vertical axis represents the mel-scaled frequency bands. The intensity of each band corresponds to the amplitude of the frequency band at a corresponding time. Figure 2.1 shows an example of visual-representation of mel-spectrogram.

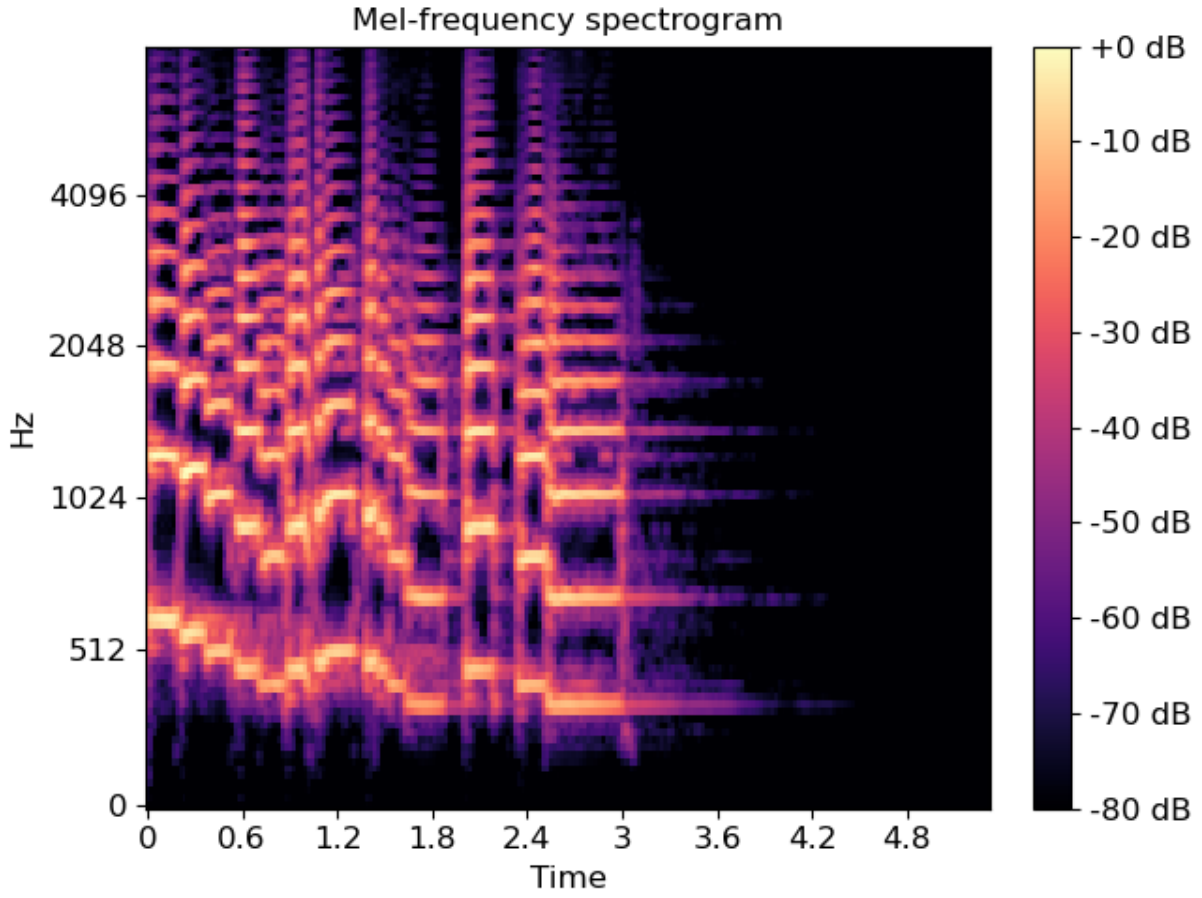

Figure 2.1 An example of mel-spectrogram from [65] with 128 mel-bands. The audio is sampled at $22kHz$, representing a maximum frequency if $8kHz$. Note that the vertical axis shows frequency in Hz and that the frequency-bands are are not linearly spaced.

2.2 Sequence-to-sequence Acoustic Model

Tacotron [119] and Tacotron 2 [102] proposed sequence-to-sequence networks for acoustic modelling with implicit alignment between the linguistic and acoustic features. The model can be trained both on raw-text and phoneme inputs, and produces mel-spectrograms as output. In this thesis all NTTS models discussed are trained on phoneme inputs.

The NTTS model described in this thesis is similar to the Tacotron 2 architecture [102]. Figure 2.2 shows the architecture of an NTTS model with encoder, decoder, and attention modules. The training and inference steps of NTTS models will be covered in chapter 4.

2.2.1 Encoder

The encoder takes the phoneme-sequence as inputs and embeds it into a 512-dimensonal embedding using a fully-connected feed-forward layer with ReLU activation [72]. The embeddings are then passed

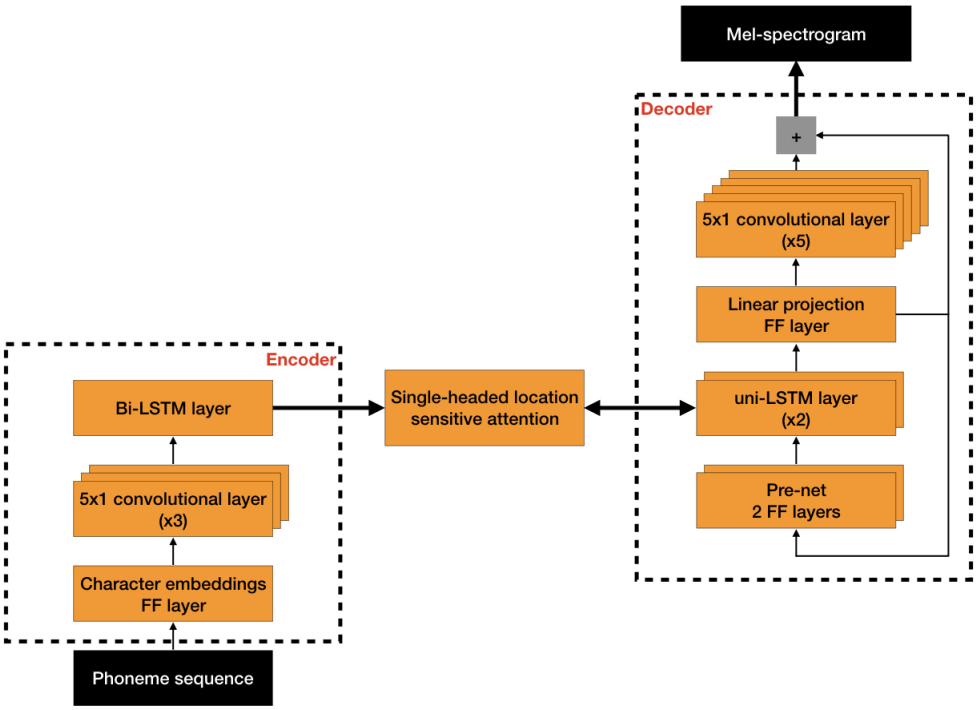

Figure 2.2 Architecture of NTTS acoustic model with encoder, decoder, and attention modules.

on to 3 layers of 2-d convolutions with a filter-size 5×1. The convolutional layers capture longer linguistic context. Batch-normalisation [36] and ReLU activation is added after each convolutional layer. The output of the convolutional layers is passed onto a single layer of bi-directional [97] LSTM [32]. The bi-directional LSTM (or bi-LSTM) layer contains 512 LSTM-cells.

2.2.2 Location-sensitive Attention

The attention module is responsible for alignment of the linguistic representation from the encoder and the acoustic representations in the decoder. Both the encoder and decoder operate on different sequence lengths. Therefore, it is important to summarise the relevant information from the encoder, for each decoder step.

Sequence-to-sequence NTTS models map an phoneme input x of length m output mel-spectrogram of y of length n. The encoder generates hidden-states of length m corresponding to the length of the input vector x:

$$\boldsymbol{h_i} = \left[\overrightarrow{h_i^T}, \overleftarrow{h_i^T}\right]^T, \quad i = 1, ..., m \tag{2.2}$$

For decoder timesteps $t = 1, ..., n$, the context vector, $\boldsymbol{c_t}$, is defined by:

$$\boldsymbol{c_t} = \sum_{i=1}^{n} \boldsymbol{\alpha_{t,i}} \cdot \boldsymbol{h_i}, \tag{2.3}$$

$$\alpha_{t,i} = Softmax(score(\boldsymbol{s_{t-1}}, \boldsymbol{h}))_i, \quad i = 1, ...m \tag{2.4}$$

$\boldsymbol{s_t}$ is the decoder hidden states at timestep t, and is a function of the previous hidden state s_{t-1}, context vector $\boldsymbol{c_t}$, and the previous output y_{t-1}:

$$\boldsymbol{s_t} = f(\boldsymbol{s_{t-1}}, \boldsymbol{c_t}, y_{t-1}) \tag{2.5}$$

The score function in Bahdanau-style additive-attention [6] is parametrised by a fully-connected feed-forward layer. The score function can formulated as:

$$score(\boldsymbol{s_{t-1}}, \boldsymbol{h_i}) = \boldsymbol{v_a}^T \, tanh(\boldsymbol{W s_{t-1}} + \boldsymbol{V h_i} + b) \tag{2.6}$$

$\boldsymbol{v_a}$, $\boldsymbol{W}$, and $\boldsymbol{V}$ are parameters learnt by the alignment model. Figure 2.3 shows a representation of a generic Bahadanau-style additive attention mechanism.

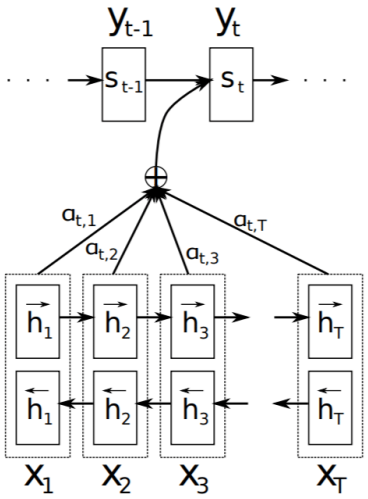

Figure 2.3 A generic framework for additive-attention mechanism [6]

Bahdanau-style attention computes attention over all encoder states. This is useful for applications like machine-translation as languages often differ syntactically and could have different word orders. Speech is inherently monotonic in nature. Looking at future timesteps for generating output at a particular timestep t is computationally expensive, and has no significant benefits. Also, it introduces potential failures where some input states could be repeated or entirely ignored.

Location-sensitive attention [18] forces the attention to be monotonic, and consistently move forward. This is done by computing the vector, $\boldsymbol{f_{t,i}}$ by convolving a matrix $\boldsymbol{F}$ over the previous alignment $\boldsymbol{\alpha_{t-1}}$:

$$f_t = \boldsymbol{F} * \boldsymbol{\alpha_{t-1}} \tag{2.7}$$

In our models, F is parametrised using 32 1-d convolutions of length 31. The vector $\boldsymbol{f_{t,i}}$ is then added to the scoring mechanism:

$$score(\boldsymbol{s_{t-1}}, \boldsymbol{h_i}) = \boldsymbol{v_a}^T \, tanh(\boldsymbol{W s_{t-1}} + \boldsymbol{V h_i} + \boldsymbol{U f_{t,i}} + b) \tag{2.8}$$

$\boldsymbol{v_a}$, $\boldsymbol{W}$, $\boldsymbol{V}$ and $\boldsymbol{U}$ are learnt parameters.

2.2.3 Decoder

The decoder in our models predicts mel-spectrograms in blocks of 5-frames at each timestep. The last frame of each timestep is passed through pre-net layers consisting of 2 fully-connected feed-forward layers with 256 units and ReLU activation. It acts as an information bottleneck and helps in learning attention alignments[102].

The output from the previous timestep processed by the pre-net, the context vector from the attention module, and the hidden-state from the previous decoder timestep is passed through uni-directional LSTM, as shown in equation 2.5. The LSTM layers in the decoder consists of 2 uni-directional LSTM layers each containing 1024 LSTM-cells.

The output of the LTSM-layer is projected through a linear fully-connected layer to predict the next set of 5 mel-spectrogram frames. The last frame from each prediction is fed-into the next decoder step as previosuly discussed. The linear layer also jointly predicts the stop-tokens during inference to indicate when the prediction has ended.

The mel-spectrogram outputs from the linear projection layer are processed through a post-net layer consisting of 5 layers of 512 convolutions with filter-size 5×1. The outputs of each layer are batch-normalised [36] and all except the last convolutional layer have $tanh$ activation. The post-net can take into context the fully-decoded sequence, and predicts a residual that is added to the final output. This has been shown to improve the overall reconstruction of the mel-spectrograms [119, 102].

2.3 Neural Vocoder

Neural vocoders are models that can convert mel-spectrogram or other acoustic features into raw-audio waveforms. During the pre-processing, we discard the phase information in the audio after STFT, focusing only on the spectral content. During the computation of mel-spectrograms, the frequencies are mapped into mel-scaled spectral bins, losing more information. This makes the inversion process non-deterministic, and the accurate reconstruction of audio waveform challenging.

Traditional methods of vocoding with mel-spectrogram inputs use Griffin-Lim algorithm [28, 83] for estimating phase information, followed by inverse STFT for audio reconstruction. [119]. However, this method produces characteristic artifacts and lower audio quality than the original waveform [102].

Neural vocoders are usually trained on waveforms sampled at $16kHZ$ or $24kHz$. Each sample is represented as a 16-bit or 24 -bit integer value. A softmax-layer would need to predict for each sample $65,536$ (2^{16}) probabilites for 16-bit audio, and ~ 16.8 million (2^{24}) probabilities for $24-bit$ audio. To make computation more efficient, ***μ*-law companding** [92] is used. μ-law companding of a normalised audio sample $-1 > x_t > 1$ is given by:

$$f(x_t) = sgn(x_t)\frac{ln(1+\mu\,|x_t|)}{ln(1+\mu)} \tag{2.9}$$

For 8-bit quantisation (256 quantised levels), $\mu = 255$.

μ-law companding is a non-linear quantisation technique. It reduces the dynamic range of speech providing better coding efficiency, and better *signal-to-noise* ratio than linear quantisation schemes.

Based on the mechanism of speech synthesis, recent advances in neural vocoders conditioned on mel-spectrograms, can be characterised into: 1) Autoregressive vocoders [113, 41], 2) GAN vocoders [54, 49, 126], 3) Flow vocoders [88, 86, 77], and 4) Diffusion vocoders [50, 15].

2.3.1 Neural Autoregressive Vocoders

Autoregressive vocoders are probabilistic models that perform sample-level predictions. These model the probability of a sample conditioned of previously generated samples.

Wavenet

Wavenet [113] was the first-attempt at a neural vocoder. Although, initially designed to be conditioned on linguistic, duration, and *f0* features, Wavenet can be modified to be conditioned on mel-spectrograms to produce high-quality audio waveforms [102]. Wavenet factorises the joint probability of the waveform, $\boldsymbol{x} = \{x_1, x_2, ..., x_t\}$, into a product of conditional probabilities:

$$p(\boldsymbol{x}) = \prod_{t=1}^{T} p(x_t | x_1, ..., x_{t-1}) \tag{2.10}$$

The conditional probabilities are modelled using stacks of dilated causal convolutions layers. The **causal convolutions** force the output at timestep, t, to be only conditioned on the previous timesteps. For 1-dimensional signals like speech waveforms, causal convolutions can be easily implemented by the asymmetric padding of the input as shown in figure 2.4. Using convolutional layer help parallelise the processing of the input.

Convolutional units with smaller filter-size suffer from the limitation of not being able to model long-term context. Increasing the filter-size risks slower training and inference due to increased number of parameters. **Dilated convolutions** [129] are effectively convolutions with larger filter-size, but with sparse filters. Having sparse filters enables the model to increase its *receptive field* and model long-term context, while keeping the number of learnable parameters low. With the depth of the model, the dilation factor (sparsity) of the convolutions is progressively increased to increase the receptive field by orders of magnitude, while keeping the number of learnable parameters low. Figure 2.4 depicts a stack of causal convolutional layers, with the dilation factor doubled at each layer.

The output of each causal convolution is passed through a gated activation unit, consisting of $tanh$ and $sigmoid$ activations. The output of the activations are then combined through element-wise multiplication. At this step, additional conditioning can be added in each layer. The additional conditioning could be speaker-embedding for multi-speaker vocoder, or upsampled linguistic features for the original Wavenet [67]. The residual from the previous layer is also added to the output of the dilated convolutional layers. This speeds up the convergence and provides stability while training deeper models [29].

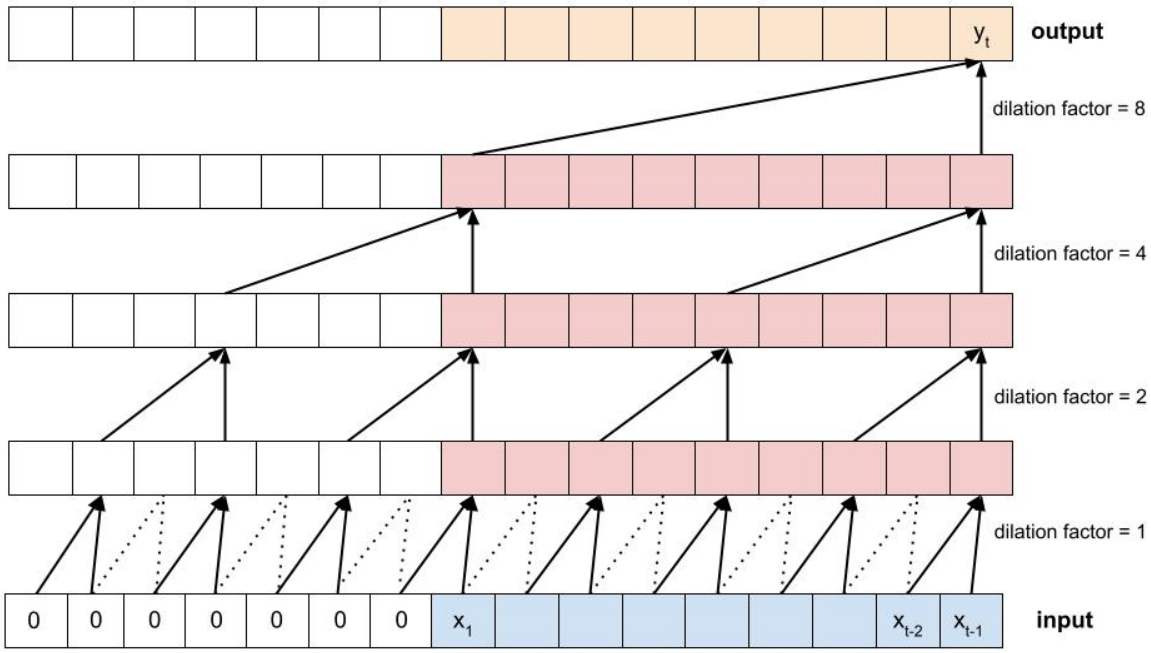

Figure 2.4 Visualisation of stacked dilated causal convolutional layers. The input is padded on the left to make the convolutions causal. The solid arrows show the receptive field at each layer for the output sample y_t.

Wavenet is trained by minimising the negative log-likelihood loss. The output can be 16-bit audio waveform, or can be μ-law quantised into 8-bit waveform. Due to its autoregressive nature, the inference from a Wavenet vocoder is slow. For modelling global context, deeper networks need to be built, increasing the total number of parameters.

To make the sampling using Wavenet faster, **Parallel Wavenet** [77] uses knowledge-distillation [31] between a larger *teacher* and a smaller *student* model. An IAF network [46] acts as a student network, which is fed with the mel-spectrogram input and noise. The output of the student network is fed to a pretrained larger Wavenet model, which acts as a teacher. KL-divergence [53] between the output distribution of the student network and the teacher network is minimised to train the student network. The student network can then be used as a standalone vocoder, which is significantly faster, but with a small loss in audio quality.

WaveRNN

WaveRNN [41] uses layers of recurrent neural networks (RNN) for modelling the sequence of audio waveforms. RNNs can inherently model long-range dependencies through their hidden states. This allows us to train shallower models resulting in faster training and inference.

The WaveRNN model consists of a conditioning network and an autoregressive network. The conditioning network takes in mel-spectrograms as input, and processes it through bidirectional Gated Recurrent Unit (GRU) [17] layers. GRU is used to model the recurrence to alleviate the problem of vanishing-gradients in RNNs. The output of the conditioning network is concatenated with the audio waveforms and input to the autoregressive network.

The input to the autoregressive network is the concatenation of the sample generated from the previous timestep, and the output of the conditioning network. The input is processed through a single layer of GRUs. The output of the GRU are then passed on to fully-connected (affine) layers with ReLU activations. The network also contains residual connections for the conditioning network at each layer. Finally, a softmax layer predicts the probability of each bit in the sample. WaveRNN can produce 16-bit audio samples 10x faster than a large Wavenet model, and with better audio quality than the student network of Parallel Wavenet [41].

The original implementation of WaveRNN proposes several optimization for faster inference and more efficient sampling. This includes a dual-softmax layer, consists of 2 8-bit softmax layers, producing 8 coarse bits, and 8 fine bits. The coarse and fine bits are concatenated to form a 16-bit audio sample. The coarse bits for the current timestep are concatenated to the output (concatenation of coarse and fine bits) from the previous timestep, to predict fine bits.The dual softmax layer reduced the output space to 8-bits instead of 16-bits resulting in more efficient prediction. Weight sparsification and batched sampling is also proposed for faster inference with lower memory footprint.

2.3.2 Universal Neural Vocoding

Wavenet and WaveRNN vocoders are trained for a single speaker, or for multiple-speakers with additional conditioning with a speaker encoding (embedding or categorical/one-hot values) for each speaker. However, neural vocoders are prone to overfitting on speaker characteristics and fail to generalise on unseen voices and speaking styles. Training neural vocoders in multi-speaker setting requires procurement of significant training data for each speaker. This makes scaling the NTTS systems expensive and time-consuming.

Universal neural vocoders are attempts towards building robust speaker and domain independent vocoders. Universal WaveRNN [61] proposed training a standard WaveRNN vocoder (without dual-softmax, and weight-sparsification optimisations) with a diverse multi-speaker, multi-lingual, and multi-style training set. They showed that exposing the WaveRNN vocoder to diverse training data helps us generalise better to unseen speech-characteristics. The study also concluded that explicitly conditioning

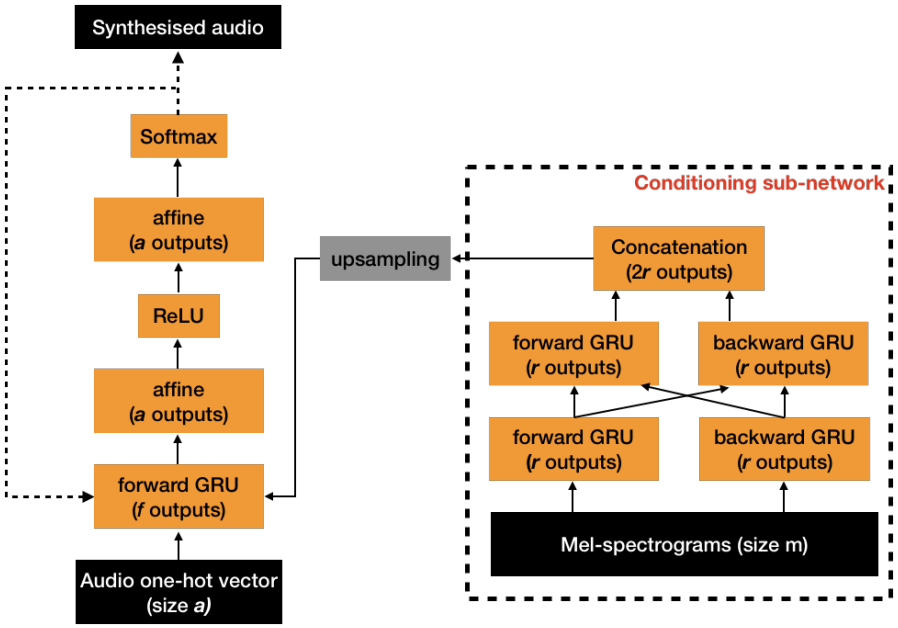

Figure 2.5 Network architecture of the Universal WaveRNN vocoder [61]

.

the vocoder with speaker and style characteristics (in the form of speaker/style embeddings) is not required.

The network architecture for the Universal WaveRNN vocoder is shown in figure 2.5. The autoregressive network consists of a single GRU layer is 896 units. This is followed by two fully-connected (affine) layers with ReLU activation between the two layers. The output of the fully-connected layers is then passed through a softmax layer with 1024 units, for synthesising 10-bit μ-law quantised waverforms at $24kHz$. The conditioning network takes in mel-spectrograms as input, and contains two bi-directional GRU-layers with 128 units. The output of the conditioning network is upsampled, and concatenated with the audio sample output from the previous timestep.

The training data for the Universal WaveRNN vocoder are sourced from different datasets. It consists of $149,134$ utterances from 74 speakers, spanning 17 languages. The training set has 22 male speakers and 52 female speakers, with approximately 2000 utterances from each speaker. The vocoder is evaluated on a variety of in-domain and out-of-domain speech scenarios. The out-of-domain scenarios consist of unseen speakers, languages, speaking styles, and non-vocal realisations (like breaths, and disfluencies).

The vocoder is shown to outperform speaker-dependant vocoders in out-of-domain speech scenarios, while maintaining similar quality in in-domain scenarios. Universal WaveRNN is not robust to noise, reverberations, and extreme energy burst (as in speech waveforms conveying excitement). However, for NTTS systems trained on studio-quality recordings, Universal WaveRNN is capable of synthesising natural-sounding waveforms. All experiments in this thesis use the Universal WaveRNN vocoder for waveform synthesis.

2.4 Evaluation

The goal of TTS systems is to accurately convey the informative and communicative element through synthesised speech. To measure these, NTTS systems are evaluated on the intelligibility and naturalness of the synthesised speech.

- **Intelligibility** is a measure of how accurately human listeners can perceive the information conveyed by the synthesised speech. Intelligibility evaluation can be automated, or through subjective human evaluations.

  Automated evaluation of TTS is done by having a pre-trained Automatic Speech Recognition (ASR) system transcribe the synthesised utterances. Word-error rate (WER) [63] between the actual transcription and the output transcriptions from the ASR system, is computed for each utterances. The mean of WER over all utterances in the evaluation set measures the intelligibility of the TTS systems.

  Subjective evaluation of intelligibility is done by presenting the human listeners with utterances that are syntactically correct, but semantically unpredictable [8]. The listeners are then asked to transcribe the utterances, based on what they hear. Mean WER is then calculated between the intended transcription and the actual transcription by human listeners.

  It is uncommon to evaluate modern NTTS systems on intelligibility as state-of-the-art NTTS systems are capable of producing high-quality waveforms with almost no gaps in intelligibility [117].

- **Naturalness** measures how close is the synthesised speech to actual human speech. Naturalness can be evaluated on several parameters like prosody, segmental quality, and appropriateness of prosody in the intended context. Measures of naturalnesss are often comparisons between the synthesised utterances and natural recordings.

  The objective metrics for naturalness measure the prosody and segmental quality between the synthesised sentences and natural recordings. These metrics are also used for monitoring the training of NTTS systems.

  Subjective evaluations of naturalness are done by presenting listeners with synthesised utterances, and asking them to rate the utterances on quality, appropriateness, preference, speaker similarity etc. Natural recordings are often provided as a reference, but are not always known to listeners.

The objective and subjective evaluation metrics used in experiments in this thesis are detailed in the following subsections.

2.4.1 Objective Metrics

Objective metrics are comparisons of the acoustic parameters of the synthesised utterances against those of natural recordings. It is possible that the sequence length of the synthesised utterance differs

from that of the corresponding natural recording. To align two, dynamic time warping (DTW) algorithm [7] is used.

Mel-spectrogram Distortion is used to assess the segmental quality of the synthesised utterances:

- **Mel-spectrogram Distortion (MSD)** [52] measures the distortion between predicted and extracted (from natural speech) mel-spectrogram coefficients and is defined as:

$$MSD = \frac{\alpha}{T}\sum_{t=1}^{T}\sqrt{\sum_{d=1}^{D-1}(c_d(t) - \hat{c}_d(t))^2} \tag{2.11}$$

$$\alpha = \frac{10\sqrt{2}}{ln10} \tag{2.12}$$

  where $c_d(t)$, $\hat{c}_d(t)$ are the d-th mel-spectrogram coefficient of the t-th frame from reference and predicted. T denotes the total number of frames in each utterance and D is the dimensionality of the mel-spectrogram coefficients. For experiments in this thesis, 80 coefficients per speech frame are used. The zeroth coefficient (overall energy) is excluded from MSD computation, as shown in equation 2.11.

For evaluating prosody, following metrics are calculated over the natural logarithm of fundamental frequency *lf0*. Since NTTS models don't explicitly predict *lf0*, it is explicitly extracted from natural recordings, and synthesised utterances for computation of the objective metrics described below.

- **F0 Root Mean Square Error (FRMSE)** is defined as:

$$FRMSE = \sqrt{\frac{\sum_{t=1}^{T}(x_t - \hat{x_t})^2}{T}} \tag{2.13}$$

  where x_t and $\hat{x_t}$ denote *lf0* extracted from reference and synthesised utterances respectively.

- **F0 Linear Correlation Coefficient (FCORR)** is the measure of the direct linear relationship between the predicted *lf0* and the reference *lf0*. It is expressed as:

$$FCORR = \frac{T\sum(x_t\hat{x_t}) - (\sum x_t)(\sum \hat{x_t})}{\sqrt{T(\sum x_t^2) - (\sum x_t)^2}\sqrt{T(\sum \hat{x_t}^2) - (\sum \hat{x_t})^2}} \tag{2.14}$$

  If x_t and $\hat{x_t}$ have a strong positive linear correlation, FCORR is close to 1.

- **Gross pitch error (GPE)** [73] is measured as percentage of voiced frames whose relative *lf0* error is more than 20%. Relative *lf0* error is defined as:

$$GPE = \frac{|x_t - \hat{x_t}|}{x_t} \times 100 \tag{2.15}$$

- **Fine pitch error (FPE)** [51] is measured as standard deviation of the distribution of relative *lf0* errors, for which relative *lf0* error is less than 20%

2.4.2 Subjective Evaluation

While objective metrics are useful for monitoring the training and for hyperparameter optimsation of the NTTS models, they usually don't correlate well with the perception of human listeners. Since end-users of the synthetic utterances generated by NTTS systems are humans, subjective evaluation through human listeners is the best estimate of the quality and usefulness of NTTS systems.

Subjective evaluations are often expensive and time-consuming to conduct, and need expert listeners to accurately assess the naturalness of output of NTTS systems. It is important for the evaluation to be diverse and with large sample sizes, for both the utterances in the evaluation set, and the listeners conducting these tests. Since the evaluations are subjective to listener's perceptions, it's also important to test the statistical significance of these evaluations [96, 121].

MOS

Mean Opinion Score (MOS) [112, 94] is the most commonly used measure for evaluating the naturalness of the synthesised utterances. The listeners are provided with one utterance per screen, and are asked to rate the utterance on naturalness. The utterances are randomised so that they aren't repeated for the same listener. The listeners rate the samples on scale ranging from 1 to 5. The recommended mapping between the listener ratings and the quality is shown in table 2.1. Before beginning the test, the listeners go through an "anchoring phase", where they are presented with examples of utterances that are expected to receive each rating. This is done in an attempt to standardise the ratings between all listeners.

MOS is an absolute measure of naturalness. A relative variation of the MOS test is comparison mean opinion score (CMOS), where the natural recording for the corresponding utterance is explicitly shown the as a reference on each screen.

MOS Rating	Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

Table 2.1 Association between ratings and quality in MOS evaluation

MUSHRA

MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) [93] presents the listeners with multiple variations of the same utterance side-by-side on a single screen. The variations of the utterance are natural recording, and the synthesised samples from different TTS systems, or are utterances in different speaking styles. The natural recording serves as the reference. The output of the baseline

systems is the anchor. Both the reference and the anchor on each screen are unknown to the listeners. The listeners are instructed to rate each utterance on a scale ranging from 1 to 100 based on its relative perceived quality. In some evaluations, the listeners are also asked to rate one of the utterances 100, to filter out any misjudgements. Since multiple systems are evaluated on a single screen, MUSHRA is a relative measure of naturalness.

As each listener simultaneously rates all the systems present in the evaluation, MUSHRA tests require fewer participants than MOS tests, and it is easier to calculate the statistical significance of through Student's t-test [81]. For comparison between multiple systems, Holm-Bonferroni correction [33] is applied.

Preference Test

Preference tests are used to compare two TTS systems. The listeners are presented with the same utterance synthesised from two different TTS systems on a single screen. The listeners choose the more natural out of the utterance A and utterance B. The order between the two systems is randomised on each screen to prevent any biases. The preference test, like MUSHRA, is a relative measure of naturalness. Preference test is also called the AB test. In some evaluation settings listeners are also provided the option, where they can indicate *'No Preference'* between the two systems. This setting is also called the ABX test. Binomial test is used to detect the statistical significance of a preference test.

2.5 Chapter Summary

In this chapter, we have provided a detailed overview of the components of Neural Text-to-speech (NTTS) system. NTTS systems combine a Sequence-to-sequence Acoustic Model and a Neural Vocoder. The Acoustic Model generates low-level acoustic features, like mel-spectrograms, given text/linguistic features as input. The low-level acoustic features are transformed into audio waveforms using Neural Vocoders, like WaveNet or WaveRNN. Universal Neural Vocoder demonstrates that training a vocoder on diverse speaker data improves generalisation.

We have also discussed the evaluation strategies for NTTS systems. NTTS systems are evaluated on their intelligibility and naturalness. NTTS systems are evaluated on their intelligibility and naturalness. Modern NTTS systems produce high-quality audio waveforms with almost no gaps in intelligibility. Naturalness is evaluated using various objective metrics and subjective evaluation. Objective metrics for evaluating naturalness are good for monitoring the training and performance of the NTTS systems, but often don't correlate well with the human perception of speech. Subjective evaluation strategies like MOS, MUSHRA, and Preference Test are used to evaluate quality, speaker similarity, context appropriateness of synthesised speech. These involve presenting samples of recorded and synthesised speech to human listeners, who are asked to rate utterances. Subjective evaluations are more accurate, but often time-consuming and expensive to conduct.

Chapter 3

Data Efficiency in Multi-speaker and Multi-style Neural Text-to-speech: An Overview

The growing popularity of speech interfaces has established the need for TTS systems to support multiple contexts - voices, speaking styles, emotions, and languages. Recording sufficient number of samples, and training independent models for each context is expensive, both in terms of time and resources. This makes scaling TTS systems in production challenging.

Data efficiency for NTTS systems is an active area of research. Several studies have proposed conditioning the acoustic model in NTTS, with additional parameters to drive the speaker-identity and speaking style of the synthesised speech. These parameters can be speaker characteristics [56, 38], or style characteristics (derived either from a reference speech sample [38], or predicted from text [120]). Another direction of research is voice-conversion [99], where the output of a speaker-dependent acoustic model is modified post-hoc with a small sample set of target speaker/style. Adaptive hyperparameter optimisation techniques for speaker adaptation [70] are also used to fine-tune a pretrained multi-speaker acoustic model with a small set of samples from the target speech characteristics. We will be exploring a few of these methods in the following sections.

3.1 Multi-speaker Training for Data Efficiency

The idea of combining data from multiple speakers comes from SPSS paradigm [125, 118]. These models work by mixing data from several speakers and training an **Average Voice Model** (AVM) [123]. During inference, the AVMs can be used to generate speech from any of the speakers in the training set, or can be fine-tuned to a new speaker with limited training data. The AVMs benefit from the quantity and diversity of data from several speakers. AVMs also help increase the robustness of the TTS system making it capable of generating speech from a wide-range of speakers and speaking styles.

Multi-speaker and multi-style NTTS models rely on external embeddings to denote the speaker or style identity. The embeddings can either be explicit, like the learnt representations from a speaker verification system [38], or be implicitly trained with the NTTS model through a one-hot speaker ID

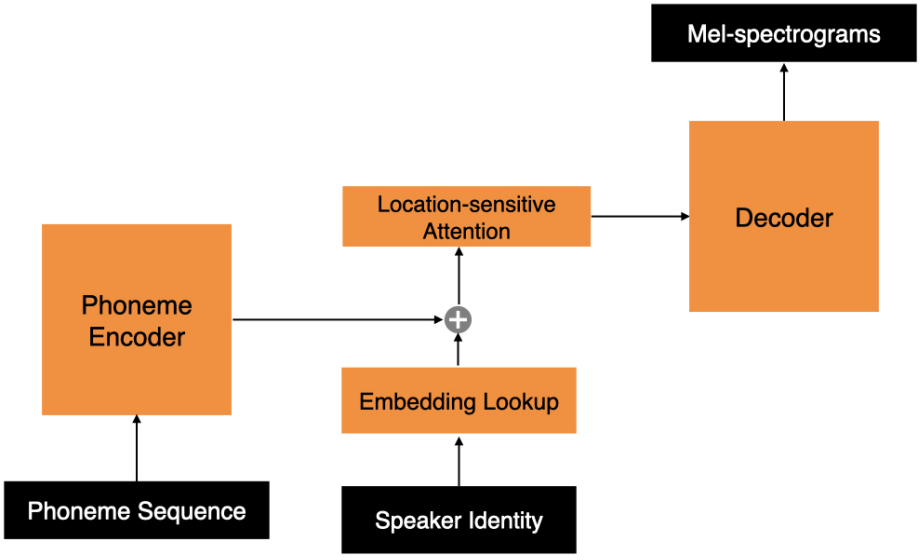

Figure 3.1 Network architecture of the multi-speaker NTTS model

.

[113, 27, 85] or a reference from mel-spectrograms [103]. However, these models are trained on large datasets containing several speakers, with comparable amount of data for each speaker.

Latorre et. al. [56] conducted extensive empirical evaluation on the amount of data required for training a speaker-dependant NTTS model. They also evaluated the proposed model with a unbalanced mixture of training data from one speaker, compared to other speakers in the training. They used the NTTS acoustic model shown in figure 2.2 in chapter 2, with a one-hot encoding for speaker identity for training the model in multi-speaker setting. The network architecture of the multi-speaker NTTS model is shown in 3.1. The model was trained on natural recordings from 7 speakers (4 female, 2 male, 1 child-like).

The study showed that a multi-speaker model trained on 5000 ($\sim$ 5 hours of data) utterances for each speaker either outperformed or matched the performance of speaker-dependant models trained on 15000 utterances each. As the size of the training data for the speaker-dependant model was increased to 25000 utterances, performance of the speaker-dependant model overtook that of the multi-speaker model.

The authors also evaluated the training in multi-speaker setting for imbalanced mixture of speakers. For this, 5000 utterances from 6 speakers were chosen. These were mixed with training data for the target speaker in two settings - with 1250 utterances (mx6+1250), with 2500 utterances (mx6+2500). The difference in performance between the two settings were insignificant. These were also compared to multi-speaker models with balanced data, trained with 2500 utterances from each speaker (mx7-2500), and with 5000 utterances from each speaker (mx7-5000). The systems mx6+2500, mx6+1250, and mx7-2500 performed similar in terms of naturalness, with insignificant difference in performance. The

two models with imbalanced training data (mx6+1250 and mx6+2500) scored 95% relative score on MUSHRA compared to mx7-5000.

The experiments concludes that NTTS models should be trained in multi-speaker setting if the training data for a target speaker is less than 15 hours. For training a model for a target speaker with limited data, at least 1250 utterances are required.

3.2 Adaptive Hyperparameter Optimisation for Few-shot Speaker Adaptation

Training multi-speaker NTTS models require re-training the NTTS model from scratch for each new target speaker. Often these trainings are expensive and still hinder scalability. **Speaker adaptation** [124] is a transfer-learning scheme, where a pre-existing model is fine-tuned for a new speaker. Through speaker adaptation techniques, a new speaker can be generated with just few minutes of training data.

Existing speaker adaptation techniques can be classified into two categories - 1) learning new speaker embedding from a pre-trained network for an auxiliary task (like speaker verification [38], or implicit speaker embedding network [71]), and 2) fine-tuning a pre-trained multi-speaker NTTS model with a small training set for a new target speaker [5, 16]. The fine-tuning approach outperforms the first in terms of naturalness, and is more flexible as it does not require modifications in the multi-speaker NTTS architecture.

3.2.1 Bayesian Optimisation for Hyperparameter Tuning

Hyperparameter tuning is the process of selecting the best configurations of a model to optimise the performance on a validation set. The most common strategies of hyperparameter optimisation are - manual search, grid search, and random search. Manual search is time-consuming, and requires manual effort from the developer. Grid search, and random search are automated strategies that select the hyperparameters from a manually pre-defined set of values. Grid-search and random-search techniques do not keep a track of performance on the previous set of parameters, and spend a major amount of time evaluating sub-optimal hyperparameters.

Bayesian Optimisation (BO) techniques [10, 100] are used to estimate the next best hyperparameter configuration at step $t+1$, by building a probabilistic model of the performance of the hyperparameter space on an objective function. This probabilistic model is called the *surrogate model*. Surrogate models are estimated from the set $\mathcal{D}_t = \{\boldsymbol{x}_i, y_i\}_{i=1,..,t}$, across the observed hyperparameter $x_i \in \mathcal{X}$ and their performance $y_i \in \mathcal{Y}$, till step t. The surrogate model is updated after each new observation of $\{\boldsymbol{x}_i, y_i\}$. Gaussian Process priors [91] are a common choice of surrogate models.

Based on the updated surrogate model, the next set of hyperparameters configurations $\boldsymbol{x}_{t+1}$ at step is chosen by an *acquisition function*. The next best set of hyperparameter configurations for a Gaussian

Process prior surrogate model is given by:

$$\boldsymbol{x}_{t+1} = \underset{\boldsymbol{x}\in\mathcal{X}}{\operatorname{argmax}}\, \mathbb{E}_{y(\boldsymbol{x})|\mathcal{D}_t}\left[\max(y'_t - y(\boldsymbol{x}), 0)|\mathcal{D}_t\right] \tag{3.1}$$

where y'_t is the best objective score in $\mathcal{Y}$ observed till step t. Equation 3.1 has a closed-form for a Gaussian Process prior, and can be computed using standard gradient-descent optimiser.

3.2.2 BOFFIN-TTS

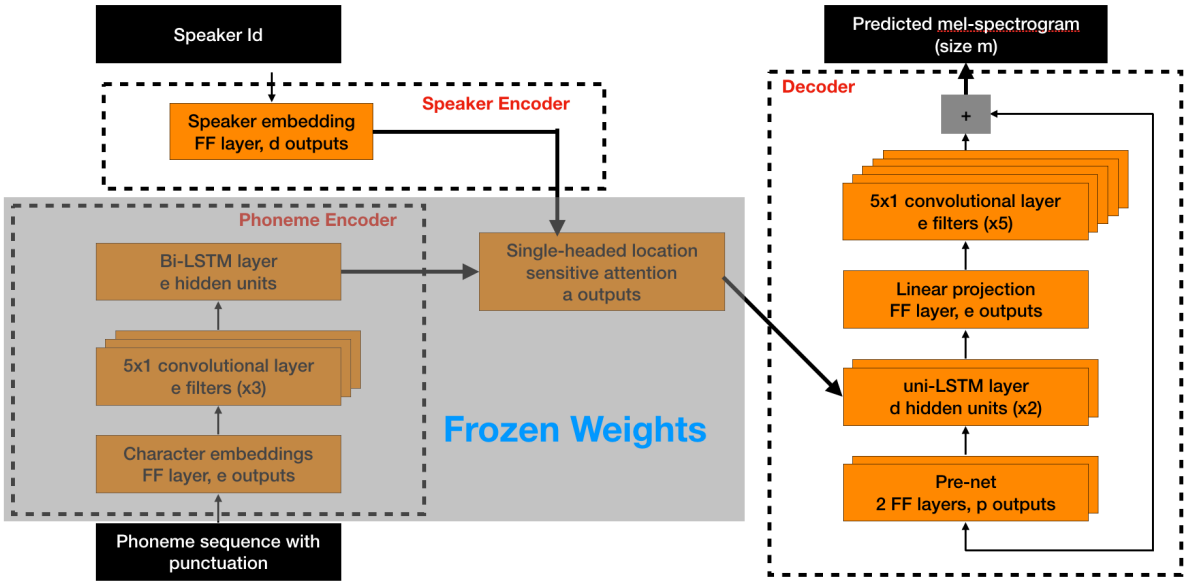

Figure 3.2 Adaptation strategy on multi-speaker NTTS proposed by BOFFIN-TTS [70]

.

BOFFIN-TTS [70] is a few-shot speaker adaption framework that uses Bayesian Optimisation to find the best hyperparameter configuration for fine-tuning a multi-speaker NTTS system for a new target speaker. The base model used in BOFFIN-TTS is a multi-speaker NTTS model discussed in section 3.1.

For the adaptation phase, the parameters of the phoneme encoder and attention module are frozen as shown in figure 3.2. This helps generalise the model better by leveraging the representations and alignment learnt from a larger training set, and prevents *catastrophic forgetting*. The parameters of the speaker embedding layer and the decoder are fine-tuned with the new utterances from the unseen target speakers.

The set of tunable hyperparameters chosen for the adaptation strategies are - {*learning rate, batch-size, decay-factor, gradient-clipping threshold, dropout, zoneout, mixing ratio for the number of utterances from the unseen speakers vs. the seen speakers, and epoch of the base model from which to begin adaptation*}. The first six are common hyperparameters that control the learning dynamics of the model.

Mixing some utterances of the seen speakers during the adaptation phase is another commonly-used strategy for preventing catastrophic forgetting. This is called the *rehearsal method*, in which samples from the base training set are also shown during fine-tuning to retain previously learnt information. The mixing ratio between utterances from the seen speaker vs. the unseen speaker is thus introduced as a hyperparameter. The fully-converged base model might need a larger number of samples for robust adaption. Fine-tuning a base model before full-convergence makes it easier to adapt with new target speakers. Hence, the starting epoch for adaptation is also introduced as a hyperparameter.

Experiments are conducted on 3 settings trained on speakers from different corpora - base model (trained on 8 high-quality speakers in the internal corpus), moderately-rich base model (trained on 14 speakers from the VCTK corpus [115], and 8 from the internal corpus), and rich-base model (trained on 200 speakers from the LibriTTS corpus [132]).

The baseline is a manually fine-tuned multi-speaker NTTS model. The base model is trained on each of the 3 settings previously discussed. 20% of the utterances from each speaker in the training data are held-out for validation, to prevent the model from overfitting. In each setting, 4 speakers are removed from the training data to be used as unseen target speakers. For the 4 unseen speakers in each setting, 100 utterances are used for speaker adaption.

For BO strategy for speaker adaptation is similar to one introduced in subsection 3.2.1. A Gaussian Process prior is fit on the configuration-evaluation pairs. The next set of hyperparameter confugration are chosen by optimising the acquisition function in equation 3.1.

The 3 training settings on different corpora are evaluated on naturalness and speaker similarity in a MUSHRA test. In the base model BO adaptation strategy improved 28% on the naturalness, and 22% on speaker similarity over the baseline. In the moderately-rich base model BO adaptation improves 57% on the speaker similarity and 13% on naturalness over the baseline. This is a more challenging setting than the base model as VCTK corpus contains larger number of speakers and highly-expressive speech samples. The rich base model is the most challenging setting as it contains a much larger number of speakers than the previous settings, and highly-expressive speech samples recorded in noisy conditions. The difference between the BO adaptation strategy and the baseline are not statistically significant both in naturalness and speaker similarity.

The experiments conclude that BO adaptation strategy for speaker adaptation can generated new target speakers with only 100 recorded samples. It outperforms manual fine-tuning in terms of speaker similarity, naturalness, and time and computation costs, especially with clean recording and less-diverse set of speakers.

3.3 Data Efficiency in Controllable NTTS Frameworks

Training or adapting NTTS models in multi-speaker setting gives us robust performance on target speakers with limited data. When these models are applied to stylistic modelling, the prosodic variations peculiar to the target styles are implicitly learnt. Often, this leads to generation of averaged prosody

of the target style during inference. The averaged prosody does not always correspond to the prosodic variations in human speech. This gap in prosodic behaviour becomes more apparent in highly-expressive or emotional speech, and in long-form content like in audiobooks.

Contrallability in the context of speech synthesis is defined as the ability to explicitly control the prosodic variations in synthetic speech through external conditioning. Disentangling prosody in from speech is a challenging problem. The prosodic variations in speech are entangled in the linguistic content [109] and the speech signal along with segmental and channel information [55]. Several attempts have been made to drive the prosodic variations in synthetic speech for a given text input through the linguistic content [87], and through a reference speech sample with similar prosodic variations [103, 120].

Skerry-Ryan et. al. [103] proposed conditioning a Tacotron 2 [102] model with reference mel-spectrograms, along with linguistic input. The representations from the reference encoder are concatenated to output linguistic representations from the phoneme encoder. During training the reference mel-spectrograms used are extracted from the recorded speech samples. This approach was extended by adding an attention layer to the reference spectrogram to learn a *code-book* of stylistic variations, called *Global Style Tokens* (GST) [120]. During inference, GST for an input sequence can be inferred either through a reference spectrogram or through text. This approach has shown great promise in same-text (parallel) prosody transfer. However, this approach fails to generate robust samples when the sequence length of reference spectrogram varies from that of the generated sample.

Conditioning an NTTS model with a reference mel-spectrogram makes it a text-conditioned autoencoder. As such, it suffers from the limitations of a traditional autoencoder. The output of the reference encoder is a latent representation of prosody. These representations are mapped on to discrete points in the latent space. It is unlikely that the latent space will cover the entire range of prosodic variations with the recorded samples. This limits us from interpolating between prosodic variations between different speech samples, and sampling from the latent space to generate unseen prosodic variations. The coverage increases with the increasing the size and diversity of the training data. As it has been already established before, procurement of larger training sets is resource-exhaustive and makes the training computationally expensive.

3.3.1 Variational Autoencoders

Variational autoencoders (VAE) [47] are probabilistic models consisting of an encoder or a *recognition model* and a decoder or the *generative model*. The encoder maps the input, $\boldsymbol{x}$, into a latent attribute $\boldsymbol{z}$. Unlike the traditional autoencoder in which the latent attribute is mapped as a discrete point in the latent space, the latent attribute in VAE is modeled as a probability distribution p_θ, parametrised by θ.

The *inference* process in a VAE refers to estimating the the posterior distribution of $p_\theta(\boldsymbol{z}|\boldsymbol{x})$. By Bayes' rule:

$$p_\theta(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_\theta(\boldsymbol{x}|\boldsymbol{z})p_\theta(\boldsymbol{z})}{p_\theta(\boldsymbol{x})} \tag{3.2}$$

The denominator of equation 3.2. also called the *evidence*, can be computed by marginalising out the latent variable z from the joint distribution $p_\theta(\boldsymbol{x}, \boldsymbol{z})$:

$$\begin{aligned} p_\theta(\boldsymbol{x}) &= p_\theta(\boldsymbol{x}, \boldsymbol{z}) dz \\ \Rightarrow p_\theta(\boldsymbol{x}) &= \int p_\theta(\boldsymbol{x}|\boldsymbol{z}) p_\theta(\boldsymbol{z}) dz \end{aligned} \tag{3.3}$$

Computing $p_\theta(\boldsymbol{x})$ is intractable in a closed form using equation 3.3. Variational inference allows us to approximate the posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ using another distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. To make sure that the approximated posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is close to the true posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$, we must minimise the KL Divergence [53] between the two distributions [1].

$$KLD(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})) = \log p_\theta(\boldsymbol{x}) + KLD(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z})) - \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) \tag{3.4}$$

Rearranging the equation 3.4, we get:

$$\log p_\theta(\boldsymbol{x}) - KLD(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x})) = \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) - KLD(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z})) \tag{3.5}$$

Since, $KLD(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}|\boldsymbol{x}))$ is intractable, and always positive, we can rewrite equation 3.5 as an inequality:

$$\log p_\theta(\boldsymbol{x}) \geq \mathbb{E}_{\boldsymbol{z}\sim q_\phi(\boldsymbol{z}|\boldsymbol{x})} \log p_\theta(\boldsymbol{x}|\boldsymbol{z}) - KLD(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z})) \tag{3.6}$$

The right-hand side of the inequality 3.6 is a lower-bound on the evidence, it's also called the **Evidence Lower Bound (ELBO)**. The evidence is the log-likelihood of generating a real data sample, which we'd like to maximize.
The negative of ELBO is used as a loss-function in VAE. The loss-function encourages the posterior distribution to stay close to the prior $p_\theta(\boldsymbol{z})$, and is a lower bound on the true log-likelihood of the data.

A standard VAE has the following components:

- **Encoder**: modelling the approximate posterior distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$, often chosen to be in the form of a multivariate Gaussian distribution with diagonal covariance:

$$\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \mu, \sigma^2 \boldsymbol{I}).$$

- **Prior**: describing the distribution $p_\theta(\boldsymbol{z})$. We assume prior to be a spherical Gaussian, $\mathcal{N}(0, \boldsymbol{I})$, as it gives us a closed-form solution to the term $KLD(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p_\theta(\boldsymbol{z}))$, and also makes the sampling process easier for generating new data points.

- **Decoder**: modelling the distribution $p_\theta(\boldsymbol{x}|\boldsymbol{z})$. The decoder, while training reconstructs the input $\boldsymbol{x}$, given the latent variable $\boldsymbol{z}$.

.

[1] The full derivation can be found in [47]

3.3.2 VAE Latent Conditioning for Controllable Speech Synthesis

The style representation of a reference speech in discrete space [103, 120], can be replaced by a probabilistic representation through a variational reference encoder [137]. This not only allows us to generate new speaking style by sampling from a continuous latent space, but also gives us the ability to interpolate between two speaking styles seen during the training to generate an unseen style or to control the intensity of the style unseen during training.

To generate the style embeddings, $\boldsymbol{z}$, mel-spectrograms from reference speech are passed through the reference encoder discussed in [103, 120]. The output of the the reference encoder is passed through two separate feed forward layers to obtain the parameters, mean ($\boldsymbol{\mu}$) and variance ($\boldsymbol{\sigma}$), of a Gaussian distributions. The variational reference encoder thus maps each input spectrogram in a probabilistic distribution given by $\mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\sigma^2}(\boldsymbol{x})\boldsymbol{I})$. The latent representation, $\boldsymbol{z}$ is then sampled from this distribution, and the concatenated to each encoder state before passing on to the attention module as input. The inference process for the latent representation, $\boldsymbol{z}$, given an input mel-spectrogram, $\boldsymbol{x}$,is described below:

$$\begin{aligned} \boldsymbol{e} &= RefEncoder(\boldsymbol{x}) \\ \boldsymbol{\mu}(\boldsymbol{x}) &= \boldsymbol{W}_{\mu}\boldsymbol{e} + b_{\mu} \\ \boldsymbol{\sigma}(\boldsymbol{x}) &= \boldsymbol{W}_{\sigma}\boldsymbol{e} + b_{\sigma} \\ \boldsymbol{z} &\sim \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}(\boldsymbol{x}), \boldsymbol{\sigma^2}(\boldsymbol{x})\boldsymbol{I}) \end{aligned} \tag{3.7}$$

The parameters $\boldsymbol{W}_{\mu}, \boldsymbol{W}_{\sigma}$ are weights of the fully-connected layers representing the parameters of the probabilistic distribution of $\boldsymbol{z}$, and b_{μ}, b_{σ} are their respective biases.

Since sampling is a stochastic process and cannot be optimised by gradient-based optimisers, the sampling process is reparametrised by introducing a random variable ϵ. The sampling process in equation 3.7 is replaced with:

$$\begin{aligned} \epsilon &\sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \\ \boldsymbol{z} &= \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\sigma^2}(\boldsymbol{x}) \odot \epsilon \end{aligned} \tag{3.8}$$

where $\odot$ represents element-wise multiplication. The sampling process thus is transferred from the parameter, $\boldsymbol{z}$ to a random variable ϵ with no trainable parameters.

The inference process of the latent variable $\boldsymbol{z}$ is shown in figure 3.3. The variational reference encoder acts as the recognition model, $q_{\phi}(\boldsymbol{z}|\boldsymbol{x})$, discussed in subsection 3.3.1. The NTTS model with a variational reference encoder, is a text-conditioned VAE that can be trained using the ELBO loss. The loss function thus becomes:

$$\mathcal{L}(\theta, \phi; \boldsymbol{x}, \boldsymbol{t}) = \beta \underbrace{KLD(q_{\phi}(\boldsymbol{z}|\boldsymbol{x})||p_{\theta}(\boldsymbol{z}))}_{\text{KL regularization term}} - \underbrace{\mathbb{E}_{\boldsymbol{z}\sim q_{\phi}(\boldsymbol{z}|\boldsymbol{x})} \log p_{\theta}(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{t})}_{\text{Reconstruction term}} \tag{3.9}$$

The input phoneme-sequence in represented by $\boldsymbol{t}$ in the equation 3.9. β is a a hyperparameter added to the KL regularisation term to prevent KL collapse [30].

For style transfer during the synthesis process, the latent variable from a reference speech sample can be used to condition the NTTS. This architecture also allows us to control the intensity of the style by

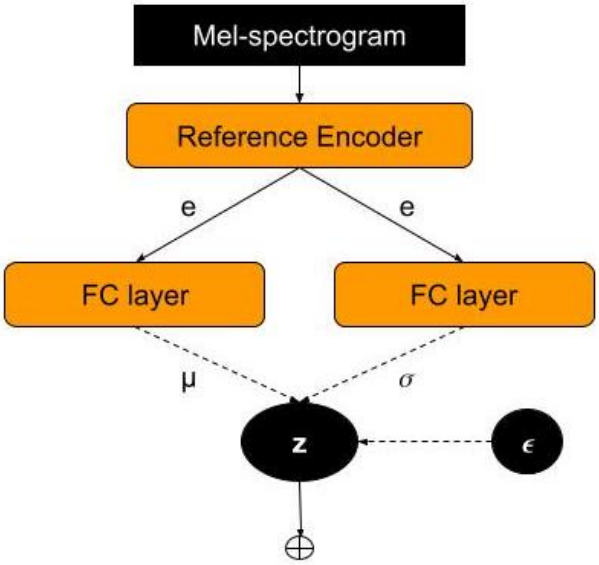

Figure 3.3 Architecture of the variational reference encoder for inference of the latent variable $\boldsymbol{z}$
.

interpolating between the latent variables from two or more reference speech samples. Since the latent space is continuous, speech in unseen styles can also be generated by sampling the latent variable from the prior $p_\theta(\boldsymbol{z})$.

3.3.3 Normalising Flows

Normalising flows, like VAEs, are likelihood-based generative models. Both VAE and normalising flows attempt to estimate the likelihood $p_\theta(\boldsymbol{x})$ of the real data. However, as discussed in equation 3.6, in the case of VAE we can only estimate the lower-bound of the likelihood of the data. Additionally, in VAE, the prior is assumed to be a standard Gaussian. This is beneficial as the Gaussian distribution has an analytical form and sampling from a Gaussian distribution is easy, The KL regularization term forces the approximate posterior to be close to the prior. In case of complex distributions with non-Gaussian features, this constraint results in poor representations, especially if the encoder is not powerful enough to learn good posterior representations.

Normalising Flows [95] use a series of invertible transformations to map a complex distribution into a simpler known distribution. In case of generative modelling and latent representation learning, this translates to mapping the likelihood $p_\theta(\boldsymbol{x})$ into a standard Gaussian distribution. This is possible with the change of variable theorem.

Let $f : \mathcal{R}^n \rightarrow \mathcal{R}^n$ be an invertible function, such that $\boldsymbol{x} = f(\boldsymbol{z})$, and $\boldsymbol{z} = f^{-1}(\boldsymbol{x})$. By change of variables theorem:

$$
\begin{aligned}
p_\theta(\boldsymbol{x}) &= p_z(\boldsymbol{z}) \left| \det \frac{\partial f(\boldsymbol{z})}{\partial \boldsymbol{x}} \right| \\
\Rightarrow p_\theta(\boldsymbol{x}) &= p_z(f^{-1}(\boldsymbol{x})) \left| \det \frac{\partial f^{-1}(\boldsymbol{x})}{\partial \boldsymbol{x}} \right|
\end{aligned}
\tag{3.10}
$$

where $\frac{\partial f(\boldsymbol{z})}{\partial \boldsymbol{x}}$ is the Jacobian matrix of the function f. We assume $p_z(\boldsymbol{z})$ to be a standard Gaussian, as in the case of VAE, to benefit from it's well-known analytical form and the ease of sampling.

A single transformation might not be expressive enough to learn more complex distributions of $p_\theta(\boldsymbol{x})$. To alleviate this, we introduce to series of K transformations, $f = f_1 \circ f_2 \circ, ..., \circ f_k$, such that

$$
\boldsymbol{x} \overset{f_1}{\longleftrightarrow} \boldsymbol{z_1} \overset{f_2}{\longleftrightarrow} \boldsymbol{z_2} .... \overset{f_k}{\longleftrightarrow} \boldsymbol{z}
$$

Each transformation gradually changes the data distribution. Using a series of transformations also allows us to use different transformation functions at each level to capture the intricacies of the data. Change of variable theorem for a sequence of transformations can be expressed as:

$$
\begin{aligned}
p_\theta(\boldsymbol{x}) &= p_z(\boldsymbol{z}) \prod_{i=1}^{K} \left| \det \frac{\partial f(\boldsymbol{z_i})}{\partial \boldsymbol{z_{i-1}}} \right|, \\
\log p_\theta(\boldsymbol{x}) &= \log p_z(\boldsymbol{z}) + \sum_{i=1}^{K} \log \left| \det \frac{\partial \boldsymbol{z_i}}{\partial \boldsymbol{z_{i-1}}} \right|
\end{aligned}
\tag{3.11}
$$

While choosing transformations f, we must adhere to following conditions:

1. The dimensions of $\boldsymbol{x}$, $\boldsymbol{z}$ and the intermediate representations $\boldsymbol{z_1}, ..., \boldsymbol{z_k}$ must be the same.
2. The function f must be invertible.
3. The computation of Jacobian determinant should be efficient and differentiable.

Normalising Flows make the computation of the exact log-likelihood of the dataset tractable, and the log-likelihood of the dataset $\mathcal{D}$ can be optimised by a gradient-based optimiser, by directly minimising the negative log-likelihood of each sample $\boldsymbol{x}$:

$$
\mathcal{L}(\mathcal{D}) = -\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \log p_\theta(\boldsymbol{x}) \tag{3.12}
$$

Based on the value of Jacobian determinant, normalising flows can be categorised into volume-preserving flows, and non-volume preserving flows. In volume-preserving flows [95, 110, 9], the distribution $p_z(\boldsymbol{z})$ has the same volume as $p_\theta(\boldsymbol{x})$ and the Jacobian determinant is equal to 1. Non-volume preserving flows [20, 21, 45] use non-linear transformations that change the volume of the data distributions. However, the transformations are still chosen such that the Jacobian determinant is efficient to compute, e.g. in affine-coupling transformation [21] the Jacobian is a lower-triangular matrix and the determinant is the product of its diagonals.

3.3.4 Style Transfer with Flexible Posterior Modelling with VAE and Normalising Flows

Normalising Flows can model the exact log-likelihood of the data. While these can act as independent density-estimation frameworks, the dimensions of the data samples, and the latent representations must be the same. This prevents us from learning low-dimension latent representations. With high-dimensional data like speech, this adds to the computation cost.

VAEs learn compact low-dimensional latent representations of the data sample. In addition to a simple prior, the approximate posterior is modelled using a normal distribution with diagonal covariance matrix. The diagonal covariance might not be flexible enough to model complex distributions.

By using normalising flows, we can transform the posterior distribution of a VAE with diagonal covariance matrix, into a distribution with full-covariance matrix. This leads to greater expressivity of the latent representations. The eigendecomposition of Any full-covariance matrix $\boldsymbol{\sigma}$ can be represented as:

$$\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T \tag{3.13}$$

where $\boldsymbol{U}$ is an orthogonal matrix containing the eigenvectors of $\boldsymbol{\Sigma}$ along its columns, and $\boldsymbol{D}$ is a diagonal matrix with the corresponding eigenvalues. By modelling the distribution of $\boldsymbol{U}$, we can use it as a linear transformation to obtain a distribution with full-covariance matrix:

$$\begin{aligned} \boldsymbol{z}_1 &= \boldsymbol{U}\boldsymbol{z}, and \\ \boldsymbol{z}_1 &\sim \mathcal{N}(\boldsymbol{U}\boldsymbol{\mu}, \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T) \end{aligned} \tag{3.14}$$

Since the value of the Jacobian determinant of an orthogonal matrix is 1, this makes the computation easier. It has been shown [110] that any orthogonal matrix of order K can be represented as a product of K Householder Transformation ($\boldsymbol{H}_1$, ..., $\boldsymbol{H}_K$). **Householder Transformation** is defined as reflection of a vector $\boldsymbol{z}_{i-1}$, along a hyperplane defined by a *Householder vector* $\boldsymbol{h}_i$ orthogonal to the hyperplane:

$$\begin{aligned} \boldsymbol{z}_i &= \boldsymbol{H}_i \boldsymbol{z}_{i-1}, \\ where, \boldsymbol{H}_i &= \boldsymbol{I} - 2\frac{\boldsymbol{h}_i \boldsymbol{h}_i^T}{||\boldsymbol{h}_i||^2} \end{aligned} \tag{3.15}$$

Householder matrices are orthogonal with Jacobian determinant equal to 1. By predicting the householder vectors $\boldsymbol{h}_i$, for each transformation, we can use the Householder transformation as an invertible transformation in flows.

In an NTTS model with variational latent conditioning, the reference encoder models the posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$. To convert this posterior distribution into a Gaussian distribution with full-covariance matrix, Householder transformations as discussed in equation 3.15 can be used [2]. At each step of the flow, the householder vectors for the corresponding step are predicted as a parameter. The householder vectors at each step are shared globally across all samples in the dataset. In [2] the best performing architecture has 16 transformations. Figure 3.4 shows the architecture of a variational reference encoder with posterior transformation using Householder flow.

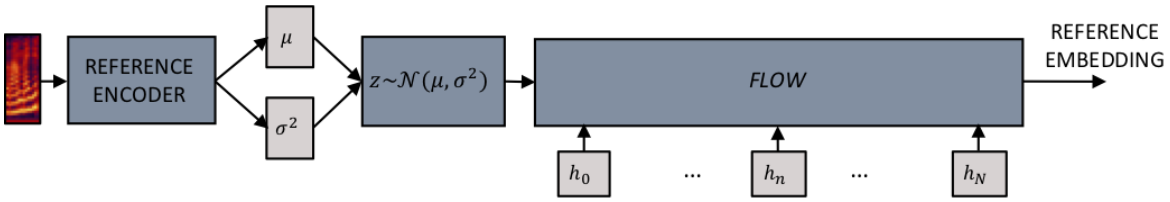

Figure 3.4 Inference of reference latent variable with VAE and Householder normalising flows [2]

The model was trained on a combination of recorded utterances from two datasets, an internal dataset (containing $\sim$ 181 hours of data from 13 speakers), and English-speakers from the VCTK [115] (containing 21 speaker, each with $\sim$ 23 minutes of speech). For style transfer, the target style is chosen from a corpus containing *'excited'* emotion. The samples in the corpus display three intensities - *low, medium, high*. One sample from each of the three intensities if chosen as the reference sample and the target for style transfer.

Posterior transformation of VAE reference encoder with normalising flows (VAE+NF) leads to improved performance on both the reconstruction loss and the KL regularisation, compared to a vanilla-VAE with no posterior transformation. Whether this translates to better naturalness and style transfer, was evaluated in a MUSHRA test against natural recordings, synthesised speech with neutral emotion, and style-transferred utterance using vanilla-VAE. The VAE+NF model performs better in both naturalness and style transfer than a vanilla-VAE. However, both these result in reduced signal quality and naturlaness score than neutral utterances. Even though significant progress has been made on controllable speech synthesis, generating high-quality samples consistently still remains a challenge.

3.4 Chapter Summary

In this chapter, we have explored techniques to used for data-efficiency in training NTTS models and adapting NTTS models to new speakers and styles.

Multi-speaker training leverages training data from several speakers to build an average voice model (AVM). Multi-speaker AVMs increase robustness and reduce per-speaker training data. Multi-speaker AVMs, however, need to be trained from scratch each time a new speaker is introduced. Speaker adaptation involves fine-tuning base AVMs to new speakers with limited training data. We discussed Bayesian Optimisation (BO) strategy to automate the hyperparameter selection for fine-tuning NTTS models. Using BO, AVMs can be adapted to a new speaker with just few minutes of training data.

AVMs generate an averaged prosody for each speaker in the training set. This produces unsatisfactory results for stylistic speech synthesis, specially in long-form context. Controllable NTTS models use probabilistic latent conditioning from a single sample of reference speech signal to drive the prosodic variations. Variational Autoencoders (VAEs) are used to encode reference speech signal as a spherical Gaussian distribution with diagonal covariance matrix, which is provided as conditioning to the NTTS

acoustic model. Encoding reference speech signals as a spherical Gaussian distribution limits expressivity. We introduced Normalising Flows that can improve the posterior flexibility by transforming the spherical Gaussian distribution into another distribution with full covariance matrix. With this, we see an improvement in both the naturalness and style-transfer over VAEs. Controllable models, although promising, show degradation in signal quality and result in lower naturalness over neutral utterances in MUSHRA evaluations.

Chapter 4

Synthesising Newscaster voice with Limited Data: A Bi-style Modelling Approach

Newscasters have a clearly identifiable dynamic style of speech. As more people are using virtual assistants, in their mobile devices and home appliances, for listening to daily news, synthesising newscaster-style of speech becomes commercially relevant. A newscaster-style of speech gives users a better experience when listening to news as compared to news generated in the neutral-style speech, which is typically used in text-to-speech synthesis. In addition, synthesising news using text-to-speech is more cost-effective and flexible than having to record new snippets of news with professional newscasters every time a new story breaks in.

Several works have explored the *controllability* of style in NTTS models through latent-variable modelling techniques [3, 34, 2]. These models not only enable us to jointly model different styles, but also allow the user to control the style through modification of latent variable during the inference. Although flexible, these models usually require a large amount of data to capture the idiosyncrasies of speaking styles. Additionally, these models are slow to train and are potentially overly complex for modelling styles of speech that are expressive, but do not display large prosodic variations. During inference, the user would need to input the latent variables to synthesise, which is not ideal for production systems.

In this chapter, we propose a model for synthesising speech in the style of a newscaster with just few hours of data. We pose this problem of generating speech in a target style with limited data, as building *'bi-style'* model that can synthesise both "neutral-style" and "newscaster style", similar to the multi-speaker modelling approach discussed in section 3.1 in chapter 3. A one-hot *style-ID* is used to differentiate between the two styles.

The contents in this chapter are published in [87]

4.1 Data Exploration

This section aims at understanding the prosodic variability in neutral-style, and newscaster-style corpora. For this purpose, we study the average variance in the natural logarithm of fundamental frequency (*lf0*) for each utterance in the two styles. The values are reported in Table 4.1. For contrast, we also study per-utterance *lf0* in a mixed-expressive corpus from the same speaker. We notice that among the three corpora, the neutral-style utterances have the lowest mean variance per utterance, making it more tractable and easier to model with NTTS than the other two corpora. Newscaster-style has a slightly higher mean variance given greater expressiveness, and the mixed-expressive corpus has the highest mean variance.

Latent-variable models [3, 34, 120, 106] tackle the problem of modelling varied expressive corpora. As we have already discussed, these models are slow to train, and require prediction or manual injection of continuous latent variables during inference. These might not be well-suited for the task of modelling newscaster-style, which even though is expressive, has much lower mean variance per utterance than the mixed-expressive corpus.

Corpus	**Variance**	**Range**
Neutral	6.32	5.66
Newscaster	6.33	5.68
Mixed expressive	6.79	5.71

Table 4.1 Analysis of mean prosodic variations based on *lf0* per utterance

Latorre et. al. [56] found that a minimum of ~ 15000 utterances (approximately 15 hours of data) are required to train a seq2seq acoustic model from scratch. Gathering 15 hours of data for each new style is both expensive and time-consuming. Given that the mean variance for the newscaster-style utterances is marginally higher than that of neutral-style utterances, we propose jointly modelling both the neutral-style and the newscaster-style, with a one-hot style ID to differentiate between the two styles. We hypothesise that the style ID will be able to effectively factorise the neutral and newscaster styles, and generate style-appropriate samples for both. This will also alleviate the problem of prediction, and injection of continuous latent variables, that might introduce additional latency in the system. During inference, the style ID can be set by modification of simple binary flags.

From our internal corpus of female US-English voice, we use ~ 20 hours of neutral-style utterances. For the newscaster-style, we use additional recordings from the same voice talent, approximating the style of American newscasters. For experiments in this paper, we use 4 hours of recorded speech for training the newscaster style. Using both these utterances to train a bi-style model provides us with enough overall data to train the acoustic model, and also help the model learn to factorise the two styles with the style ID input.

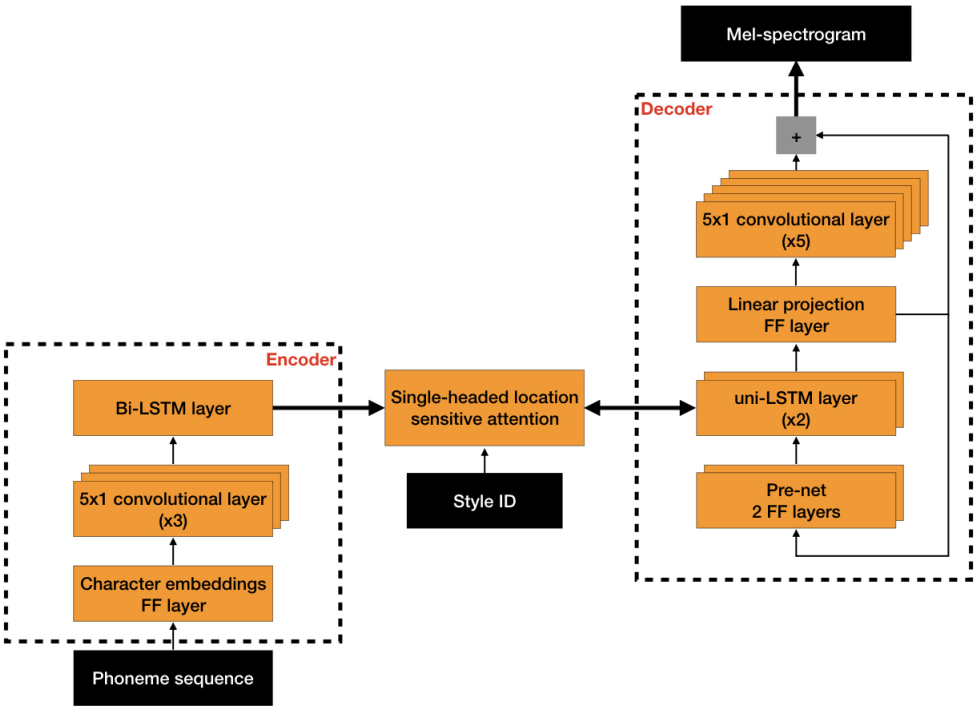

Figure 4.1 Architecture of the bi-style NTTS acoustic model

.

4.2 Model Description

Our proposed model is composed of two modules - an NTTS Acoustic Model and a Waveform Synthesis model. The NTTS acoustic model takes phonemes as inputs, and predicts temporal acoustic features, e.g. mel-spectrograms. The predicted acoustic features are then converted to time-domain audio waveforms by the Waveform Generation module.

4.2.1 NTTS Acoustic Model

The NTTS acoustic model consists of the phoneme encoder, style ID input, a single-headed location-sensitive attention block, and the decoder module.

The style ID is a two-dimensional one-hot vector (representing whether the input utterance belongs is in the neutral-style or newscaster-style), which is projected into continuous space by an embedding lookup layer to produce a *style embedding*. The style embedding is concatenated at each step of the output of the phoneme encoder.

Single-headed location-sensitive attention [18] is applied to the concatenated outputs. A unidirectional LSTM-layer takes the concatenated vector of the output vector of the attention block and the pre-net layer as an input.

The decoder, in each step, predicts blocks of 5 frames of 80-dimensional mel-spectrograms. We define a frame as a $50ms$ sequence, with an overlap of $12.5ms$. The last frame of the previous outputs is passed to the pre-net layer as input for generating the next set of frames. The architecture of the NTTS acoustic model is shown in figure 4.1

4.2.2 Waveform Generation

We use the pre-trained Universal WaveRNN discussed in subsection 2.3.2 in chapter 2 to convert the mel-spectrograms predicted by our context generation module into high-fidelity audio waveforms.

4.3 Experimental Protocol

The news stories are on an average longer than neutral-style utterances, and consist of multiple sentences. Seq2seq models have a tendency to lose attention and have misalignment in longer input sequences during inference. To alleviate this, we split the news stories into individual sentences in both the training and the test sets. Splitting into individual sentences also enables us to train the model on larger batch size, helping the model to converge faster and with lesser perturbation of the training loss. To convert the utterances into phoneme sequences, we use our internal G2P tool, which encodes the phonemes, stress marks, and punctuations as one-hot vectors.

4.3.1 Training

We train the model using an L1 loss in the decoder output for mel-spectrogram prediction. To indicate when to stop predicting the decoder outputs, we have a linear stop token generator at the decoder outputs, trained jointly with the context generation module. The stop token generator is trained with an L2 loss. During training, the stop token is linearly increased from 0 at the beginning of the sentence to 1 at the end.

ADAM optimizer [44] is used to minimise the training loss, with learning rate decay. The model is trained with teacher-forcing on the decoder outputs. The attention weights are normalised to add up to 1 using a softmax layer.

The decoder is trained with dropout [105] regularisation (with probability 0.1). No dropout is used in the encoder module. We use mel-spectrogram distortion [52] to monitor the input-output alignment, and the training loss to get a rough estimation on the convergence of our model. We also synthesise some held-out sentences to monitor the segmental quality and the prosody of our system, as the perceptual quality of the generated samples does not always align with the lower training and validation losses, and spectrogram distortion metrics.

4.3.2 Evaluation

We use the objective metrics discussed in subsection 2.4.1 for evaluating the proposed model. We compare the newscaster style generated by our bi-style NTTS model (**News Bi-style**) against a neutral NTTS model (**Neutral**), and a concatenative TTS system (**Concatenative**). The neutral NTTS model is trained in a single-speaker setting. The concatenative TTS systems is a hybrid unit-selection systems, trained on neutral style of speech, driven by state-level parametric predictions, as described in [48].

We compare acoustic parameters extracted from the synthesised sentences, and the natural recordings for the analysis of prosody and segmental quality. To match the predicted sequence length to the reference sequence length for all comparisons, we use the dynamic time warping (DTW) algorithm [7]. We use Mel-spectrogram Distortion to assess the segmental quality of the synthesised sentences.

We compare the models on prosody and segmental quality. We use the same text-prompts for generating speech with all the models in evaluation. All the systems use the voice of the same female speaker.

4.4 Results and Discussions

	Segmental Quality	**Prosody**			
System	**MSD (dB)**	**FRMSE (Hz)**	**FCORR**	**GPE (%)**	**FPE (cents)**
Concatenative	6.07	44.85	0.28	33.58	5.68
Neutral	5.27	44.81	0.30	32.02	5.63
News Bi-style	**4.52**	**42.90**	**0.35**	**28.89**	**5.57**

Table 4.2 Objective metrics for analysis of prosody and segmental quality. High FCORR indicates better prosody. For all other metrics, lower value indicates better performance.

The scores of the objective evaluation are shown in table 4.2. The concatenative TTS system produces characteristic artefacts at the points of concatenation, that affect the segmental quality. Through NTTS modelling we can generate high-quality audio samples without the concatenation artefacts.

We have already discussed in section 3.3 that the prosodic variation are embedded in speech signal along with segmental information, and it's hard to entirely disentangle prosody with segmental quality. In terms of prosody, the concatenative and neutral NTTS samples have similar performance, with differences driven mostly by the segmental quality of the synthesised speech samples.

Our proposed model (News Bi-style) obtains consistently better scores in both prosody and segmental quality than neutral NTTS and concatenative TTS systems. This shows that high-quality audio samples for a new speaking style can be generated with limited data for the target style, using bi-style modelling with neutral data for the same speaker.

Additionally, we also show that using one-hot conditioning, we can effectively factorise the two speaking styles. This shows great promise for the scalability of NTTS models in production, where a

single multi-style model can be deployed to generate speech in several speaking styles. Bi-style training approach also provides additional regularisation with varied training data, that improves the segmental quality over single-speaker neutral NTTS system.

The objective metrics although do not always correlate well with human perception of speech, the give us a good sense of initial progress without additional costs of setting up subjective evaluation.

Through manual inspection of generated newscaster-style samples from held-out data, we observe that even though we are able to capture some prosodic variations of the newscaster style of speech, the model still produces an averaged prosody of the target style. This affects the appropriateness of the style in some samples. Further work is needed to control the prosodic variations in newscaster style, to suit the context of the text prompts used for synthesis.

4.5 Chapter Summary

In this chapter, we adapt multi-speaker AVMs to propose a bi-style modelling approach to synthesise newscaster style utterances with only 4 hours of stylistic training data.

We combine the newscaster style data with a large corpus of neutral utterances from the same speaker, in a single NTTS model with one-hot style conditioning. Through object metrics, we show that the bi-style model outperforms both neutral NTTS and concatenative TTS in both prosody and segmental quality. We also show that the model effectively factorises the two styles through only one-hot style conditioning without the need for a reference speech signal from the target style.

Chapter 5

Improving Naturalness with Contextualised Word Embeddings

Prosody in spoken language is closely tied to both by the syntax and semantics of the linguistic content being verbalised [116, 109]. Syntactic elements like word order and phrase boundaries in a sentence affect the rhythm and intonation in speech. Semantics of a sentence relate to the emotional information, stress pattern, and overall tone of the speech. The prosody of spoken content can also help disambiguate homographs and homonyms. Thus, informing a TTS system with the syntax and semantics of the linguistic content is important for natural-sounding speech synthesis.

Conventional NTTS acoustic models use a single encoder for linguistic inputs (phonemes/character embeddings). With no explicit front-end processing of the text input, the phoneme encoder cannot be solely relied upon to capture higher-level text characteristics like syntax or semantics.

Recent advances in representation learning for text [84, 19] have allowed us to come up with contextualised linguistic representations that not only capture the syntax and semantics of a word, but also the linguistic context of the word as a function of the entire sentence. These contextualised representations can be effective conditioning information for long-form TTS content, as seen in news and audiobooks.

In this chapter, we propose conditioning the bi-style NTTS acoustic model discussed in chapter 4 with contextualised word embeddings (CWE), by introducing an additional encoder. The CWE encoder provides additional linguistic context to the NTTS acoustic model without the need for explicit hand-crafted front-end features.

5.1 Contextualised Word Embeddings

Word embeddings are dense continuous representations of words that capture the meaning of the words and the context in which they are present. Traditional word embedding methods [68, 82] learn these representations from large unlabelled corpora. These methods generate embeddings only for words seen during training, and only a single embedding for each word is generated regardless of the different contexts in which the words can be present (*polysemy*). Later methods alleviate some of these

The contents in this chapter are published in [87]

shortcoming by either learning representations for subword unit [12, 40], or learning separate embedding for each word sense [74].

Contextualised word embeddings (CWE) methods generate dynamic word embeddings for each word, which is a function of the overall input sentence. CWE methods use sequence-learning architectures [32, 114] to encode the context around pivot words through unsupervised language modelling [84, 19, 89, 90] or through an supervised downstream task like machine translation [64]. These embeddings are transferable on a range of downstream NLP tasks and applicable to any language with large corpora.

5.1.1 Embeddings from Language Model

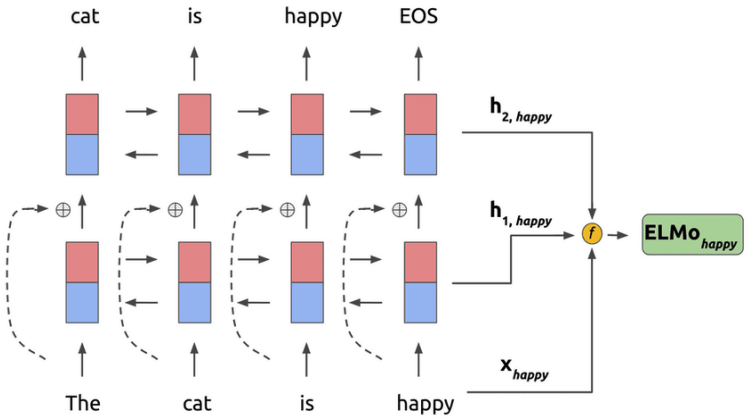

Figure 5.1 Visual representation of ELMo. From [79]

.

Embedding from Language Model (ELMo) [84] is a CWE method that uses unsupervised pretraining via language modelling to generate contexualised embeddings for downstream tasks.

The input text in ELMo is processed through a character-level convolutional layer that captures the subword information of the input. This is beneficial for representing morphological variations of the words, and for generating CWE for out-of-vocabulary words. The output of the character-level convolutional layer is passed to two bi-directional LSTM layers for capturing the context in which the word is presented. This makes the CWE a function of the sentence. Each layer learns progressively higher-level information of the word. It has been shown that the lower layers capture the syntactic information, while the final layer captures the semantic information of the word. The final representation is a linear combination of the outputs of the character-level convolutional layer, the first bidirectional-LSTM layer, and the final bidirectional-LSTM layer. The architecture of ELMo is shown in figure 5.1

Using CWE generated from ELMo, the authors beat state-of-the art in six downstream NLP tasks - question answering, textual entailment, semantic role labelling, coreference resolution, named-entity recognition, and sentiment analysis.

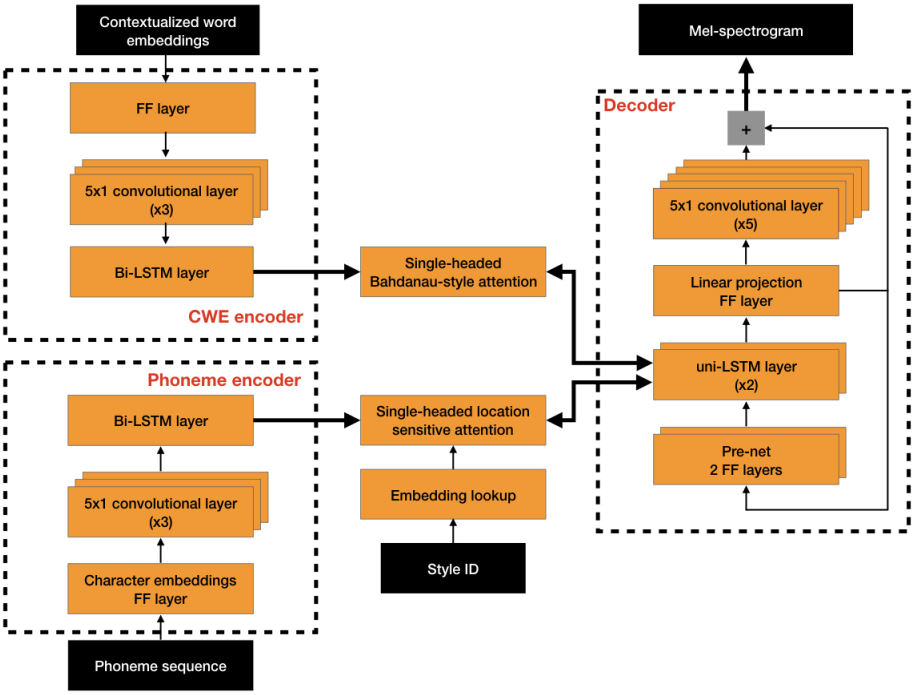

Figure 5.2 Architecture of the bi-style NTTS acoustic model with CWE Encoder

.

5.2 Model Description

We introduce a CWE encoder as additional conditioning to the bi-style NTTS acoustic model described in chapter 4. For each sentence in the training set we extract ELMo features using publicly available CLI tool [26]. This model is pre-trained on the 1 Billion Word Benchmark dataset [14]. During training these features are fed to CWE encoder. CWE encoder has a similar topology to that of the phoneme encoder.

Encoded ELMo embeddings are passed to the decoder through Bahdanau-style attention [6]. For CWE encoder, we choose Bahdanau-style attention instead of the location-sensitive attentions as the syntax and semantic information in the input text isn't necessarily monotonic. We hypothesise that this can help the decoder consider broader context. The attention module of the CWE encoder operates independently of the location-sensitive attention for the phoneme encodings.

The NTTS acoustic model, conditioned on the phoneme representations and the CWE of the input text, generates frame-level mel-spectrograms. The proposed architecture, shown in figure 5.2, is a multi-scale conditioning on the input text, focusing both on phoneme-level (through phoneme encoder) and word-level (through CWE encoder) contexts. These spectrograms are then converted into raw audio waveform using the Universal WaveRNN vocoder.

System	Description
Concatenative	Concatenative-based unit selection system driven by state-level statistical parametric predictions
Neutral	Neutral-style NTTS speech
News w/o CWE	Newscaster-style NTTS speech without CWE conditioning
News with CWE	Newscaster-style NTTS speech with CWE conditioning
Recordings	Natural speech waveforms

Table 5.1 Systems present in the MUSHRA evaluation

5.3 Experimental Protocol

The training setup for the proposed model is similar to that of the bi-style NTTS. The model is trained on the neutral and newscaster style utterances, with each news story split into individual sentences.

We use L1 loss for the decoder output, and L2 loss for the stop-token prediction. The decoder is trained using teacher-forcing of the mel-spectrogram. ADAM optimiser with learning rate decay is used to minimise the training loss.

5.3.1 Evaluation

Objective Metrics

The objective metrics used for evaluating the proposed model are discussed in subsection 2.4.1 in chapter 2. We compare the bi-style newscaster NTTS model with CWE conditioning (**News with CWE**) against the bi-style newscaster NTTS model proposed in chapter 4 (**News w/o CWE**), neutral NTTS model (**Neutral**), and a neutral-style concatenative TTS system (**Concatenative**). The models are evaluated on their prosody and segmental quality.

Subjective Evaluation

We conduct additional subjective evaluations with human listeners and consider these as the final outcome of our experiments. We concatenate the synthesised news-style sentences into full news stories, to capture the overall experience of our intended use-case. Each utterance is 3-5 sentences long, and the average duration is 33.47 seconds.

We test our system with 10 expert listeners with native linguistic proficiency in English, using the MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) methodology [93]. The systems used in this evaluation are described in Table 5.1. The listeners are asked to rate the appropriateness of each system as a newscaster voice on a scale of 0 to 100. For each utterance, 5 stimuli are presented to the listeners side-by-side on the same screen, representing the 5 test systems in a random order. Each listener rates 51 screens.

5.4 Results and Discussions

5.4.1 Objective Metrics

	Segmental Quality	Prosody			
System	**MSD (dB)**	**FRMSE (Hz)**	**FCORR**	**GPE (%)**	**FPE (cents)**
Concatenative	6.07	44.85	0.28	33.58	5.68
Neutral	5.27	44.81	0.30	32.02	5.63
News w/o CWE	**4.52**	42.90	0.35	28.89	5.57
News with CWE	4.54	**42.14**	**0.36**	**27.59**	**5.55**

Table 5.2 Objective metrics for analysis of prosody and segmental quality. High FCORR indicates better prosody. For all other metrics, lower value indicates better performance.

The scores for the objective metrics are shown in Table 5.2. We observe that both of our newscaster-style models obtain consistently better scores on all metrics, than neutral NTTS and concatenative-based system. Furthermore, we also observe that conditioning the newscaster-style model with CWE helps improve the prosody of the synthesised utterances.
There's a slight loss in segmental quality when conditioning the model with CWE, but it appears to be imperceptible to human listeners in the MUSHRA test.

5.4.2 Subjective Evaluation

System	Mean score	Median score	Mean Rank	Median Rank
Concatenative	28.31	21.5	4.60	5
Neutral	42.44	37.0	3.86	4
News w/o CWE	68.15	76.0	2.67	3
News with CWE	72.4	80.0	2.41	2
Recordings	91.61	100.0	1.45	1

Table 5.3 Listener ratings from the MUSHRA evaluation

The listener responses from the subjective evaluation are shown in Figure 5.3. In Table 5.3 the descriptive statistics for the MUSHRA evaluation are reported. The proposed model closes the gap between concatenative-based synthesis for newsreading, which is still largely the industry standard, and the natural recordings by 69.7%. The gap compared with the neutral NTTS voice is also closed by 60.9%. All of the systems present in the MUSHRA test are statistically significant from each other at a p-value of 0.01. This significance is observed across the listener responses using a t-test. Holm-Bonferroni correction was applied due to the number of condition pairs to compare. This significance is

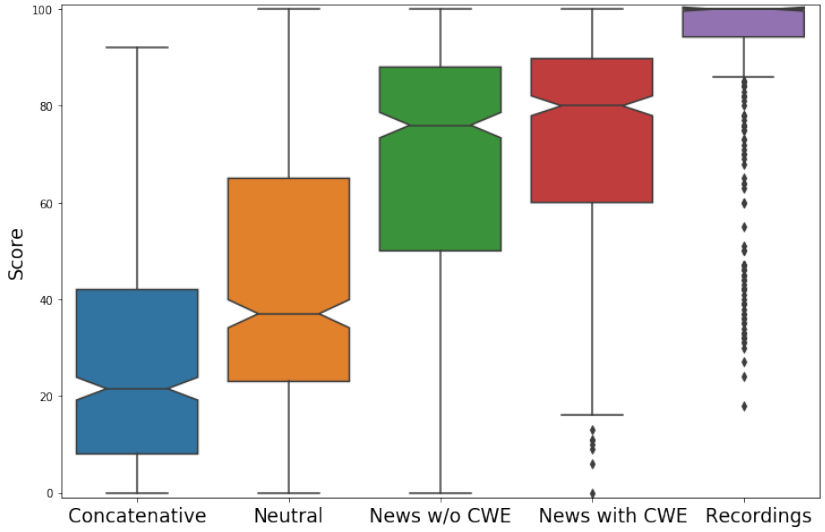

Figure 5.3 Boxplot of the listener responses in the MUSHRA evaluation

also observed over the MUSHRA responses in terms of the rank order awarded by listeners. For this a Wilcoxon signed-rank test applying Holm-Bonferroni correction was used.

The concatenative-based system is prone to audible artefacts at the concatenation-points, primarily due to abrupt changes in fundamental frequency in voiced phonemes. This reduces the perceived naturalness of synthesised speech. The neutral-style system is unable to model the prosody that is distinct to the newscaster-style of speech. A higher score for the newscaster-style model with CWE conditioning with respect to the model without, provides evidence supporting the hypothesis that CWE features help model the prosodic variation better given the additional information on the syntactic and semantic contexts of words in the sentence.

We also generated a violin plot (Figure 5.4) depicting the distribution of the rank-order awarded to the systems in the test. We notice that for some of the utterances, the listeners have ranked our newsreader voice (both with and without CWE) higher than the natural recordings, showing that our proposed models are able to closely mimic the recordings in terms of prosody and naturalness.

5.4.3 Effect of Contextualised Word Embeddings on Prosody Modelling

To further reinforce the effect of CWE on prosody modelling for newscaster-style, a preference test was conducted comparing newscaster-style with and without CWE conditioning, using 10 expert listeners. Listeners were informed to rate the systems in terms of their naturalness, and were asked to choose between News with CWE, News w/o CWE, or indicate *No Preference*(NP).

The listener responses are shown in Table 5.4. The samples conditioned on contextual word embeddings are shown to be significantly preferred (43.2%) over the samples generated without (31%), with $p < 0.01$. A binomial test was used to detect statistical significance.

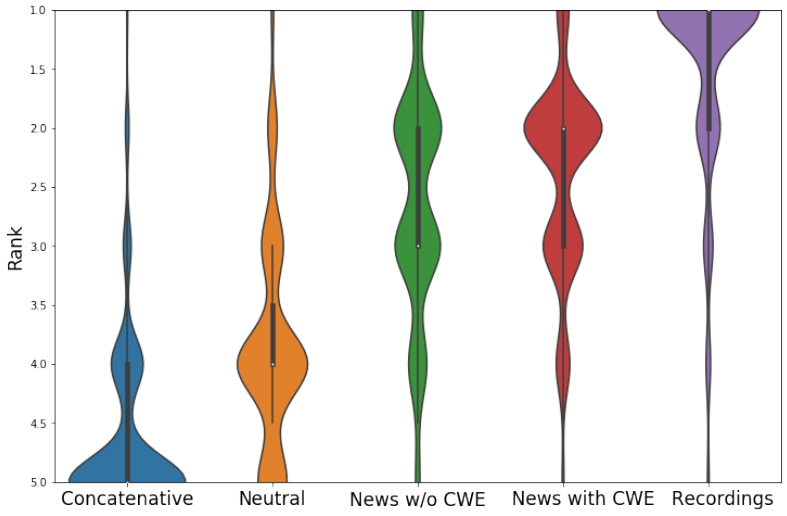

Figure 5.4 Violin plot of the rank-order awarded by listeners

Preference	No. of Votes	% Votes
News with CWE	259	43.2%
News w/o CWE	186	31%
No Preference	155	25.8%

Table 5.4 Preference test between systems with and without CWE conditioning

5.4.4 Analysis of Speech Tempo

System	Neutral	Newscaster
Recordings	11.63	14.02
with CWE	10.12	13.88
w/o CWE	10.11	13.65

Table 5.5 Speech tempo: recordings vs test systems

We define speech tempo of a corpus as the average number of phonemes present per second. Speech tempo is a crucial aspect in differentiating between the neutral and the newscaster styles. The newscaster-style is more dynamic than the neutral-style utterances, with higher speech tempo. In Table 5.5 we report the speech tempo in the neutral-style, and the newscaster-style for natural recordings, and compare those with our models with and without CWE. We observe that the model conditioned on CWE can better model the speech tempo in both styles. This gives us additional evidence that conditioning the model on CWE helps us synthesise samples that are not only more style-appropriate, but are also better in naturalness with respect to natural recordings.

5.5 Chapter Summary

In this chapter, we discuss the relationship between the prosody of speech and syntax and semantics of the text. We showed that naturalness in NTTS models can be improved by conditioning it on linguistic context beyond phonemes.

We used pre-trained Contextualised Word Embedding (CWEs) to extract syntactic and semantic context from text. We introduced a CWE Encoder with Bahdanau attention as additional conditioning the the bi-style NTTS model discussed in the previous chapter.

Objective metrics showed that CWE conditioning improves prosody of newscaster style utterances, with similar performance on neutral style utterance as the baseline model. MUSHRA evaluation for naturalness indicates that CWE conditioning closes the gap between the baseline model and recordings by 60.9%. Bi-style model with CWE conditioning was also rated higher in context-appropriateness in Preference Test.

Chapter 6

Conclusions and Future Directions

In this thesis, we studied the components of a Neural Text-to-speech (NTTS) system and highlighted the challenges of training an independent NTTS model for a new speaker and style. NTTS systems can generate high-quality audio waveforms for a given text input. These implicitly learn the relationship between the text and acoustic parameters without needing multi-stage front-end and back-end processing, or requiring any hand-crafted features. However, vast amounts of training data is required for training NTTS model compared to traditional concatenative and SPSS systems. This affects the scalability of NTTS systems to new speakers and styles, specially the ones with limited data.

We studied the data-efficiency techniques for scaling NTTS systems to new speakers and style. We first studied training NTTS models in multi-speaker setting. Combining data from several speakers gives the model the volume of data required to learn the text and acoustic alignments, while only requiring fractional amount of data for each speaker compared to training a speaker-independent systems. With a single multi-speaker in production, several voices can be generated. Although, for introducing a new speaker, the model needs to be retrained. We discussed Bayesian Optimisation (BO) techniques for adapting a multi-speaker model to new speakers. BO techniques automatically find the best set of hyperparameters for fine-tuning a pre-trained multi-speaker NTTS model to introduce a new speaker just few minutes of training data, without degrading the naturalness of the synthesised speech for pre-existing speakers. We also studied using variational autoencoders and normalising flows for one-shot transfer of speaker and style characteristics. These methods make the synthesis in a new speaking style controllable. However, these come with significant degradation to naturalness and the signal quality of the synthesised speech. These also require selection of appropriate reference audio for the injection of the target style.

We propose a bi-style NTTS model for synthesising speech in newscaster style with one-fourth the amount of data required for training a speaker and style-dependent model. We combine training data for the newscaster style with neutral style recordings from the same speaker. We use a one-hot style-ID to differentiate between the two styles. The proposed model outperforms concatenative TTS system in the naturalness of the synthesised speech for both neutral and newscaster style utterances. The synthesised newscaster style utterances also have wider prosodic range than neutral utterances. Through objective

metrics, we show that a new target style can be generated with limited training data in a multi-style setting. Further, the styles can be factorised with a single one-hot style-ID.

Finally, we propose conditioning the bi-style NTTS model for neutral and newscaster style, with contextualised word embeddings (CWE) for each utterance. The CWE conditioning gives the NTTS acoustic model additional linguistic context (both syntactic and semantic) for generating more natural sounding speech. This also provides the NTTS model with multi-scale (phoneme-level and word-level) information of the input text. We conduct extensive objective and subjective evaluations for measuring the effect of CWE conditioning. With CWE conditioning, we significantly improve the prosody modelling ability, and thus naturalness, of the NTTS acoustic model.

6.1 Future Directions

Through multi-style modelling we can effectively generate stylistic speech with limited data. However, the lack of controllability in these systems still remains a challenge. Latent variable conditioning requires selection of appropriate reference speech samples for style transfer. Recently, dynamic selection of reference based on syntactic distance between the input text and the reference text has been proposed [111]. We also saw a degradation is naturalness and signal quality in controllable speech synthesis frameworks. There has been initial developments in the robustness of such models [34, 43], but their performance in multi-style setting, especially with limited data is still an open problem.

Conditioning NTTS model with pre-trained CWE has shown promising results in improving naturalness and prosody modelling, in our experiments. However, unsupervised pre-training might not give us the most appropriate representations for prosody modelling. Further research is needed in joint-training or fine-tuning CWEs for guiding the prosodic variations in stylistic speech synthesis.

Computing CWEs through sequential models is a slow process and hinders real-time speech synthesis, thus limiting their usefulness in production and streaming systems. A distilled version of BERT [19] uses a 40% smaller model to produce CWEs that has shown comparable performance in NLP tasks that use CWEs generated by a larger BERT [19] model, a slight degradation in performance. Using distilled CWEs might help bring the synthesis time for NTTS with CWE conditioning closer to real-time.

Sequence-to-sequence acoustic models tend to lose attention over longer input sequences, making alignment over longer text inputs difficult. Thus, long-form content is synthesised one sentence at a time. A limitation that arises from this is not being able inform the model with broader prosodic context. CWE only provide word-level and sentence-level context. Broader text representations need to be designed that provide the acoustic model with paragraph-level or document-level context, which we hypothesise will help further improve prosody-modelling in long-form content.

Publications

Related Publications

1. **Prateek, N.**, Łajszczak, M., Barra-Chicote, R., Drugman, T., Lorenzo-Trueba, J., Merritt, T., Ronanki, S. and Wood, T., 2019. In Other News: A Bi-style Text-to-speech Model for Synthesizing Newscaster Voice with Limited Data. In Proceedings of NAACL-HLT (pp. 205-213).

2. Merritt, T.E., Nadolski, A.F., **Prateek, N.**, Putrycz, B., Chicote, R.B., Aggarwal, V. and Breen, A.P., Amazon Technologies Inc, 2020. *Text-to-speech (TTS) processing*. U.S. Patent 10,692,484.

Other Publications

1. Aggarwal, V., Cotescu, M., **Prateek, N.**, Lorenzo-Trueba, J. and Barra-Chicote, R., 2020, May. Using vaes and normalizing flows for one-shot text-to-speech synthesis of expressive speech. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6179-6183). IEEE.

2. Moss, H.B., Aggarwal, V., **Prateek, N.**, González, J. and Barra-Chicote, R., 2020, May. Boffin tts: Few-shot speaker adaptation by bayesian optimization. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 7639-7643). IEEE.

3. Aggarwal, V., **Prateek, N.**, Chicote, R.B., and Breen, A.P., Amazon. Technologies Inc., *Synthetic Speech Processing*. U.S. Patent 11,017,763.

4. Chicote, R.B., Aggarwal, V., Breen, A.P., Hernandez, J.G. and **Prateek, N.**, Amazon Technologies Inc, 2022. *Text-to-speech processing using input voice characteristic data*. U.S. Patent 11,373,633.

Bibliography

[1] S. Achanta, A. Antony, L. Golipour, J. Li, T. Raitio, R. Rasipuram, F. Rossi, J. Shi, J. Upadhyay, D. Winarsky, et al. On-device neural speech synthesis. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1155–1161. IEEE, 2021.

[2] V. Aggarwal, M. Cotescu, N. Prateek, J. Lorenzo-Trueba, and R. Barra-Chicote. Using vaes and normalizing flows for one-shot text-to-speech synthesis of expressive speech. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6179–6183. IEEE, 2020.

[3] K. Akuzawa, Y. Iwasawa, and Y. Matsuo. Expressive speech synthesis via modeling expressions with variational autoencoder. *arXiv preprint arXiv:1804.02135*, 2018.

[4] J. A. Argente. From speech to speaking styles. *Speech communication*, 11(4-5):325–335, 1992.

[5] S. Arik, J. Chen, K. Peng, W. Ping, and Y. Zhou. Neural voice cloning with a few samples. *Advances in neural information processing systems*, 31, 2018.

[6] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[7] R. Bellman and R. Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2):1–9, 1959.

[8] C. Benoît, M. Grice, and V. Hazan. The sus test: A method for the assessment of text-to-speech synthesis intelligibility using semantically unpredictable sentences. *Speech communication*, 18(4):381–392, 1996.

[9] R. v. d. Berg, L. Hasenclever, J. M. Tomczak, and M. Welling. Sylvester normalizing flows for variational inference. *arXiv preprint arXiv:1803.05649*, 2018.

[10] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24, 2011.

[11] A. W. Black, H. Zen, and K. Tokuda. Statistical parametric speech synthesis. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–1229. IEEE, 2007.

[12] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146, 2017.

[13] Z. Cai, Y. Yang, and M. Li. Cross-lingual multi-speaker speech synthesis with limited bilingual training data. *Computer Speech & Language*, 77:101427, 2023.

[14] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

[15] N. Chen, Y. Zhang, H. Zen, R. J. Weiss, M. Norouzi, and W. Chan. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.

[16] Y. Chen, Y. Assael, B. Shillingford, D. Budden, S. Reed, H. Zen, Q. Wang, L. C. Cobo, A. Trask, B. Laurie, et al. Sample efficient adaptive text-to-speech. *arXiv preprint arXiv:1809.10460*, 2018.

[17] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[18] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015.

[19] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[20] L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[21] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[22] D. M. Eberhard, G. F. Simons, and C. D. Fennig. *Ethnologue: Languages of the World*. SIL International, Dallas, Texas, 26th edition, 2023.

[23] M. Ephratt. Linguistic, paralinguistic and extralinguistic speech and silence. *Journal of pragmatics*, 43(9):2286–2307, 2011.

[24] M. Eskénazi. Changing speech styles: Strategies in read speech and casual and careful spontaneous speech. 1992.

[25] M. Eskenazi. Trends in speaking styles research. In *Third European Conference on Speech Communication and Technology*, 1993.

[26] M. Gardner, J. Grus, M. Neumann, O. Tafjord, P. Dasigi, N. F. Liu, M. Peters, M. Schmitz, and L. S. Zettlemoyer. AllenNLP: A deep semantic natural language processing platform. In *ACL workshop for NLP Open Source Software*, 2018.

[27] A. Gibiansky, S. Arik, G. Diamos, J. Miller, K. Peng, W. Ping, J. Raiman, and Y. Zhou. Deep voice 2: Multi-speaker neural text-to-speech. *Advances in neural information processing systems*, 30, 2017.

[28] D. Griffin and J. Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243, 1984.

[29] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[30] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

[31] G. Hinton, O. Vinyals, J. Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015.

[32] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[33] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[34] W.-N. Hsu, Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Y. Wang, Y. Cao, Y. Jia, Z. Chen, J. Shen, et al. Hierarchical generative modeling for controllable speech synthesis. *arXiv preprint arXiv:1810.07217*, 2018.

[35] A. J. Hunt and A. W. 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*, volume 1, pages 373–376. IEEE, 1996.

[36] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[37] A. Javaloy and G. García-Mateos. Text normalization using encoder–decoder networks based on the causal feature extractor. *Applied Sciences*, 10(13):4551, 2020.

[38] Y. Jia, Y. Zhang, R. J. Weiss, Q. Wang, J. Shen, F. Ren, Z. Chen, P. Nguyen, R. Pang, I. L. Moreno, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *arXiv preprint arXiv:1806.04558*, 2018.

[39] Z. Jin, A. Finkelstein, G. J. Mysore, and J. Lu. Fftnet: A real-time speaker-dependent neural vocoder. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 2251–2255. IEEE, 2018.

[40] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016.

[41] N. Kalchbrenner, E. Elsen, K. Simonyan, S. Noury, N. Casagrande, E. Lockhart, F. Stimberg, A. Oord, S. Dieleman, and K. Kavukcuoglu. Efficient neural audio synthesis. In *International Conference on Machine Learning*, pages 2410–2419. PMLR, 2018.

[42] H. Kawahara. Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds. *Acoustical science and technology*, 27(6):349–353, 2006.

[43] J. Kim, S. Kim, J. Kong, and S. Yoon. Glow-tts: A generative flow for text-to-speech via monotonic alignment search. *Advances in Neural Information Processing Systems*, 33:8067–8077, 2020.

[44] D. P. Kingma and J. L. Ba. Adam: Amethod for stochastic optimization. In *Proc. 3rd Int. Conf. Learn. Representations*, 2014.

[45] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[46] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29, 2016.

[47] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[48] V. Klimkov, A. Nadolski, A. Moinet, B. Putrycz, R. Barra-Chicote, T. Merritt, and T. Drugman. Phrase break prediction for long-form reading tts: Exploiting text structure information. 2018.

[49] J. Kong, J. Kim, and J. Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033, 2020.

[50] Z. Kong, W. Ping, J. Huang, K. Zhao, and B. Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

[51] D. A. Krubsack and R. J. Niederjohn. An autocorrelation pitch detector and voicing decision with confidence measures developed for noise-corrupted speech. *IEEE Transactions on signal processing*, 39(2):319–329, 1991.

[52] R. Kubichek. Mel-cepstral distance measure for objective speech quality assessment. In *Proceedings of IEEE pacific rim conference on communications computers and signal processing*, volume 1, pages 125–128. IEEE, 1993.

[53] S. Kullback and R. A. Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[54] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Z. Teoh, J. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. *Advances in neural information processing systems*, 32, 2019.

[55] D. R. Ladd. *Intonational phonology*. Cambridge University Press, 2008.

[56] J. Latorre, J. Lachowicz, J. Lorenzo-Trueba, T. Merritt, T. Drugman, S. Ronanki, and V. Klimkov. Effect of data reduction on sequence-to-sequence neural tts. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7075–7079. IEEE, 2019.

[57] J. Laver and L. John. *Principles of phonetics*. Cambridge university press, 1994.

[58] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu. Neural speech synthesis with transformer network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6706–6713, 2019.

[59] N. Li, S. Liu, Y. Liu, S. Zhao, M. Liu, and M. Zhou. Close to human quality tts with transformer. *arXiv preprint arXiv:1809.08895*, 2018.

[60] J. Llisterri. Speaking styles in speech research. In *Workshop on Integrating Speech and Natural Language*. Citeseer, 1992.

[61] J. Lorenzo-Trueba, T. Drugman, J. Latorre, T. Merritt, B. Putrycz, and R. Barra-Chicote. Robust universal neural vocoding. *arXiv preprint arXiv:1811.06292*, 2018.

[62] J. Lyons. *Semantics: Volume 2*, volume 2. Cambridge university press, 1977.

[63] A. Marzal and E. Vidal. Computation of normalized edit distance and applications. *IEEE transactions on pattern analysis and machine intelligence*, 15(9):926–932, 1993.

[64] B. McCann, J. Bradbury, C. Xiong, and R. Socher. Learned in translation: Contextualized word vectors. *Advances in neural information processing systems*, 30, 2017.

[65] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 2015.

[66] T. Merritt, R. A. Clark, Z. Wu, J. Yamagishi, and S. King. Deep neural network-guided unit selection synthesis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5145–5149. IEEE, 2016.

[67] T. E. Merritt, A. F. Nadolski, N. Prateek, B. Putrycz, R. B. Chicote, V. Aggarwal, and A. P. Breen. Text-to-speech (tts) processing, June 23 2020. US Patent 10,692,484.

[68] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[69] M. Morise, F. Yokomori, and K. Ozawa. World: a vocoder-based high-quality speech synthesis system for real-time applications. *IEICE TRANSACTIONS on Information and Systems*, 99(7):1877–1884, 2016.

[70] H. B. Moss, V. Aggarwal, N. Prateek, J. González, and R. Barra-Chicote. Boffin tts: Few-shot speaker adaptation by bayesian optimization. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7639–7643. IEEE, 2020.

[71] E. Nachmani, A. Polyak, Y. Taigman, and L. Wolf. Fitting new speakers based on a short untranscribed sample. In *International Conference on Machine Learning*, pages 3683–3691. PMLR, 2018.

[72] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Icml*, 2010.

[73] T. Nakatani, S. Amano, T. Irino, K. Ishizuka, and T. Kondo. A method for fundamental frequency estimation and voicing decision: Application to infant utterances recorded in real acoustical environments. *Speech Communication*, 50(3):203–214, 2008.

[74] A. Neelakantan, J. Shankar, A. Passos, and A. McCallum. Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*, 2015.

[75] T. Nekvinda and O. Dušek. One model, many languages: Meta-learning for multilingual text-to-speech. *arXiv preprint arXiv:2008.00768*, 2020.

[76] H. J. Nussbaumer. The fast fourier transform. In *Fast Fourier Transform and Convolution Algorithms*, pages 80–111. Springer, 1981.

[77] A. Oord, Y. Li, I. Babuschkin, K. Simonyan, O. Vinyals, K. Kavukcuoglu, G. Driessche, E. Lockhart, L. Cobo, F. Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*, pages 3918–3926. PMLR, 2018.

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

[79] P. Opuchlich. Comparing unsupervised word embedding models to build a large scale text recommendation system. 10 2019.

[80] D. O'Shaughnessy. Human and machine. *Speech Communication, Addison-W esley Publishing Company*, pages 41–127, 1997.

[81] K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, 185:71–110, 1894.

[82] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[83] N. Perraudin, P. Balazs, and P. L. Søndergaard. A fast griffin-lim algorithm. In *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*, pages 1–4. IEEE, 2013.

[84] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[85] W. Ping, K. Peng, A. Gibiansky, S. O. Arik, A. Kannan, S. Narang, J. Raiman, and J. Miller. Deep voice 3: 2000-speaker neural text-to-speech. *arXiv preprint arXiv:1710.07654*, 2017.

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

[87] N. Prateek, M. Łajszczak, R. Barra-Chicote, T. Drugman, J. Lorenzo-Trueba, T. Merritt, S. Ronanki, and T. Wood. In other news: A bi-style text-to-speech model for synthesizing newscaster voice with limited data. *arXiv preprint arXiv:1904.02790*, 2019.

[88] 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 (ICASSP)*, pages 3617–3621. IEEE, 2019.

[89] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[90] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[91] C. E. Rasmussen. *Gaussian processes in machine learning*. Springer, 2004.

[92] C. Recommendation. Pulse code modulation (pcm) of voice frequencies. In *ITU*, 1988.

[93] I. Recommendation. Method for the subjective assessment of intermediate sound quality (mushra). *ITU, BS*, pages 1543–1, 2001.

[94] I. Recommendation. Itu-t p. 800.1,". *Mean Opinion Score (MOS) Terminology*, 2003.

[95] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[96] J. Roy, J. Cole, and T. Mahrt. Individual differences and patterns of convergence in prosody perception. *Laboratory Phonology*, 8(1), 2017.

[97] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[98] S. R. Schweinberger, H. Kawahara, A. P. Simpson, V. G. Skuk, and R. Zäske. Speaker perception. *Wiley Interdisciplinary Reviews: Cognitive Science*, 5(1):15–25, 2014.

[99] J. Serrà, S. Pascual, and C. Segura Perales. Blow: a single-scale hyperconditioned flow for non-parallel raw-audio voice conversion. *Advances in Neural Information Processing Systems*, 32, 2019.

[100] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2015.

[101] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[102] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[103] R. Skerry-Ryan, E. Battenberg, Y. Xiao, Y. Wang, D. Stanton, J. Shor, R. Weiss, R. Clark, and R. A. Saurous. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *international conference on machine learning*, pages 4693–4702. PMLR, 2018.

[104] J. Sotelo, S. Mehri, K. Kumar, J. F. Santos, K. Kastner, A. Courville, and Y. Bengio. Char2wav: End-to-end speech synthesis. 2017.

[105] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[106] D. Stanton, Y. Wang, and R. Skerry-Ryan. Predicting expressive speaking style from text in end-to-end speech synthesis. *arXiv preprint arXiv:1808.01410*, 2018.

[107] S. S. Stevens, J. Volkmann, and E. B. Newman. A scale for the measurement of the psychological magnitude pitch. *The journal of the acoustical society of america*, 8(3):185–190, 1937.

[108] J. Taylor and K. Richmond. Analysis of pronunciation learning in end-to-end speech synthesis. In *INTERSPEECH*, pages 2070–2074, 2019.

[109] P. Taylor. *Text-to-speech synthesis*, pages 111–112. Cambridge university press, 2009.

[110] J. M. Tomczak and M. Welling. Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

[111] S. Tyagi, M. Nicolis, J. Rohnke, T. Drugman, and J. Lorenzo-Trueba. Dynamic prosody generation for speech synthesis using linguistics-driven acoustic embedding selection. *arXiv preprint arXiv:1912.00955*, 2019.

[112] I. T. Union. Methods for subjective determination of transmission quality, itu-t recommendation.

[113] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. In *SSW*, page 125, 2016.

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

[115] C. Veaux, J. Yamagishi, K. MacDonald, et al. Superseded-cstr vctk corpus: English multi-speaker corpus for cstr voice cloning toolkit. 2016.

[116] M. Wagner and D. G. Watson. Experimental and theoretical advances in prosody: A review. *Language and cognitive processes*, 25(7-9):905–945, 2010.

[117] P. Wagner, J. Beskow, S. Betz, J. Edlund, J. Gustafson, G. Eje Henter, S. Le Maguer, Z. Malisz, E. Szekely, C. Tånnander, et al. Speech synthesis evaluation—state-of-the-art assessment and suggestion for a novel research program. In *Proceedings of the 10th Speech Synthesis Workshop (SSW10)*, 2019.

[118] V. Wan, J. Latorre, K. Yanagisawa, N. Braunschweiler, L. Chen, M. J. Gales, and M. Akamine. Building hmm-tts voices on diverse data. *IEEE Journal of Selected Topics in Signal Processing*, 8(2):296–306, 2013.

[119] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, et al. Tacotron: A fully end-to-end text-to-speech synthesis model. corr abs/1703.10135, 2017.

[120] Y. Wang, D. Stanton, Y. Zhang, R. Skerry-Ryan, E. Battenberg, J. Shor, Y. Xiao, F. Ren, Y. Jia, and R. A. Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *arXiv preprint arXiv:1803.09017*, 2018.

[121] M. Wester, C. Valentini-Botinhao, and G. E. Henter. Are we using enough listeners? no!—an empirically-supported critique of interspeech 2014 tts evaluations. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[122] Z. Wu, O. Watts, and S. King. Merlin: An open source neural network speech synthesis system. In *SSW*, pages 202–207, 2016.

[123] J. Yamagishi. Average-voice-based speech synthesis. *Tokyo Institute of Technology*, 2006.

[124] J. Yamagishi, T. Kobayashi, Y. Nakano, K. Ogata, and J. Isogai. Analysis of speaker adaptation algorithms for hmm-based speech synthesis and a constrained smaplr adaptation algorithm. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(1):66–83, 2009.

[125] J. Yamagishi, B. Usabaev, S. King, O. Watts, J. Dines, J. Tian, R. Hu, Y. Guan, K. Oura, K. Tokuda, et al. Thousands of voices for hmm-based speech synthesis. In *Tenth Annual Conference of the International Speech Communication Association*, 2009.

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

[127] Z.-J. Yan, Y. Qian, and F. K. Soong. Rich-context unit selection (rus) approach to high quality tts. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4798–4801. IEEE, 2010.

[128] S. Yolchuyeva, G. Németh, and B. Gyires-Tóth. Text normalization with convolutional neural networks. *International Journal of Speech Technology*, 21(3):589–600, 2018.

[129] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015.

[130] H. Ze, A. Senior, and M. Schuster. Statistical parametric speech synthesis using deep neural networks. In *2013 ieee international conference on acoustics, speech and signal processing*, pages 7962–7966. IEEE, 2013.

[131] H. Zen, Y. Agiomyrgiannakis, N. Egberts, F. Henderson, and P. Szczepaniak. Fast, compact, and high quality lstm-rnn based statistical parametric speech synthesizers for mobile devices. *arXiv preprint arXiv:1606.06061*, 2016.

[132] H. Zen, V. Dang, R. Clark, Y. Zhang, R. J. Weiss, Y. Jia, Z. Chen, and Y. Wu. Libritts: A corpus derived from librispeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

[133] H. Zen and H. Sak. Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4470–4474. IEEE, 2015.

[134] H. Zen, K. Tokuda, and A. W. Black. Statistical parametric speech synthesis. *speech communication*, 51(11):1039–1064, 2009.

[135] H. Zhang, R. Sproat, A. H. Ng, F. Stahlberg, X. Peng, K. Gorman, and B. Roark. Neural models of text normalization for speech applications. *Computational Linguistics*, 45(2):293–337, 2019.

[136] Y. Zhang, R. J. Weiss, H. Zen, Y. Wu, Z. Chen, R. Skerry-Ryan, Y. Jia, A. Rosenberg, and B. Ramabhadran. Learning to speak fluently in a foreign language: Multilingual speech synthesis and cross-language voice cloning. *arXiv preprint arXiv:1907.04448*, 2019.

[137] Y.-J. Zhang, S. Pan, L. He, and Z.-H. Ling. Learning latent representations for style control and transfer in end-to-end speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6945–6949. IEEE, 2019.