UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Expressivity transfer in deep learning based text-to-speech synthesis

THÈSE

présentée et soutenue publiquement le 07 Juillet 2022

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Ajinkya Kulkarni

Composition du jury

Président :	Axel Roebel	Directeur de Recherches - IRCAM
Rapporteurs :	Yannis Stylianou	Professeur - University of Crete
	Damien Lolive	Maitre de Conférences HDR - Université de Rennes 1
Examinatrice :	Marie Tahon	Maitre de Conférences - Université du Mans
Encadrants :	Denis Jouvet	Directeur de Recherches - Inria
	Vincent Colotte	Maitre de Conférences - Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Résumé

Bien que la synthèse de parole à partir du texte ait connu ces dernières années un immense succès dans le domaine de l'interaction homme-machine, les systèmes actuels sont perçus comme monotones en raison de l'absence d'expressivité. L'expressivité dans la parole réfère généralement aux caractéristiques suprasegmentales représentées par les émotions, les styles d'expression, les gestes et expressions faciales, etc. Une synthèse vocale expressive devrait permettre d'améliorer considérablement l'expérience utilisateur avec les machines. Le développement d'un système de synthèse de parole expressive dépend fortement des données vocales disponibles. Cette thèse vise à développer un système de synthèse de parole expressive dans la voix d'un locuteur pour lequel seules des données vocales neutres sont disponibles. L'objectif principal de la thèse est d'étudier des approches d'apprentissage profond pour explorer le désenchevêtrement des informations locuteur et d'expressivité dans un contexte de synthèse de parole multilocuteur. Le contexte d'application concerne l'expressivité en tant qu'émotion avec des classes d'émotion bien définies.

Nous proposons différentes architectures de réseaux neuronaux profonds pour créer des représentations latentes du locuteur et de l'expressivité dans des configurations de synthèse de parole multilocuteurs. Pour le transfert de l'expressivité, les représentations de l'expressivité et du locuteur sont utilisées pour synthétiser la parole expressive dans la voix du locuteur souhaité. Nous utilisons également le critère *multiclass N-Pair loss* lors de l'apprentissage pour améliorer la représentation latente de l'expressivité (meilleure séparation des émotions dans l'espace latent), ce qui permet d'améliorer le transfert d'expressivité. Nous étudions également les modèles génératifs profonds permettant une modélisation tractable et évolutive de données vocales complexes et hautement dimensionnelles, ces modèles étant reconnus pour une synthèse vocale de haute qualité. Nous avons enrichi ces modèles pour étudier leur capacité de transfert d'expressivité.

L'évaluation des systèmes proposés est difficile car aucune donnée de référence de parole expressive n'est disponible dans la voix du locuteur cible. Par conséquent, nous proposons deux mesures d'évaluation subjectives, le MOS expressivité et le MOS locuteur, qui indiquent les performances de transfert de l'expressivité et de rétention de la voix du locuteur cible. Nous proposons également une métrique d'évaluation objective basée sur la similarité en cosinus pour mesurer la pertinence de l'expressivité et de la voix du locuteur.

Les résultats obtenus démontrent la capacité des approches proposées à transférer l'expressivité tout en maintenant la qualité globale de la parole expressive synthétisée dans la voix du locuteur cible. Cependant, l'identification des paramètres des réseaux neuronaux représentant explicitement les attributs des caractéristiques du locuteur et de l'expressivité reste difficile. Les caractéristiques d'expressivité et de locuteur sont des aspects conjoints de la prosodie.

Mots-clés: Synthèse de parole, Expressivité, Apprentissage profond, Apprentissage par transfert

Abstract

Recently, text-to-speech (TTS) synthesis has gained immense success in the human-computer interaction domain. Current TTS systems are monotonous due to the absence of expressivity. Expressivity in speech generally refers to suprasegmental speech characteristics represented by emotions, speaking styles, and the relationship between speech and gestures, facial expressions, etc. It seems likely that expressive speech synthesis provides the ability to improve the user experience with machines greatly. The development of an expressive TTS system heavily relies on the speech data used in training the system. The thesis aims at developing an expressive TTS system in a speaker's voice for which only neutral speech data is available. The main focus of the thesis is to investigate deep learning approaches for exploring the disentanglement of speaker information and expressivity in a multispeaker TTS setting. The scope of the work incorporates expressivity as an emotion attribute with well-defined emotion classes.

We present various deep neural network architectures to create latent representations of speaker and expressivity in multispeaker TTS settings. During the expressivity transfer phase, representations from expressivity and speaker are used to interpolate for synthesizing expressive speech in desired speaker's voice. We present a deep metric learning framework for improving the latent representation of expressivity in a multispeaker TTS system setting, which results in improved expressivity transfer. The thesis work also investigates the expressivity transfer capability of probability density estimation based on deep generative models. The usage of deep generative models provides scalable modeling of complex, high-dimensional speech data and tractability of the system, resulting in high-quality speech synthesis.

The evaluation of the proposed systems is a challenging aspect of the thesis, as no reference expressive speech data was available in the target speaker's voice. Therefore, we propose two subjective evaluation metrics, speaker MOS and expressive MOS, which indicate the performance of the framework to transfer the expressivity and the retention of the target speaker's voice. As it is a time-consuming process to conduct a subjective evaluation each time system is developed, we propose a cosine similarity-based evaluation metric to measure the strength of expressivity and the speaker's voice.

The obtained results demonstrate the ability of the proposed work to transfer the expressivity with maintaining the overall quality of synthesized expressive speech in the target speaker's voice. It is hard to identify which neural network parameters represent the attributes of speaker characteristics and expressivity. Moreover, expressivity and speaker characteristics are bounded aspects of prosody parameters.

Keywords: Text-to-speech, Expressivity, Deep learning, Transfer learning

Acknowledgments

The journey of my Ph.D. thesis research is naturally accompanied by acknowledgments to everyone who has assisted me, collaborated with me, and advised me in many ways.

First and foremost, I would like to thank my supervisors, Denis Jouvet and Vincent Colotte, for believing in me and giving me the opportunity to conduct the doctoral research. The guidance and the freedom to explore various deep learning architectures that they offered were essential factors for thesis work. Moreover, my thesis supervisor, Denis Jouvet, provided continual support in every step of doctoral research as well as activities outside the scope of thesis work, such as teaching, supervision of interns, and collaboration with other teams. Furthermore, I would like to thank both supervisors for their detailed feedback on the thesis manuscript.

I would like to say a big thank you to Miguel Couceiro for providing me the opportunity to teach a software development course and co-supervise an internship project. The scientific discussions with Miguel were always joyful and enlightening to carry forward good research work. Overall experience with Miguel on various collaborative projects equipped me to apply the essence of machine learning in various modalities other than speech processing. After that, I would like to express my gratitude to Emmanuel Vincent for allowing me to assist him in teaching PyTorch labs for a deep learning course. Furthermore, I am thankful to Emmanuel Vincent for involving me in discussions with the Inria startup studio.

I would like to state special thanks to my colleague and friend Ioannis Douros for all the support and help during all this time. Additionally, I would like to thank all the members of the Multispeech group. In particular, thanks to Sandipana Dowerah, Md Sahid, Imran Sheikh, Seyed Ahmad Hosseini, and Théo Biasutto-Lervat. Also, I am thankful to Esteban Marquer for always assisting me with the French language for administrative work. I would like to express sincere gratitude towards all the administrative staff members for their support in the documentation and administrative work, which includes Helene Cavallini, Annick Jacquot, Sylvie Hilbert, Delphine Hubert, Souad Boutaguermouchet, and Sabrina Ferry.

I would like to state heartfelt thanks to my friends, Lázaro Calles Calles, Nitesh Kulkarni, Paul Ricketts, Sanket Zarkar, and Nishchal Nandanwar, for always being there for me and making my stay in Europe happier and more effortless. Finally, I would like to say sincere gratitude to my parents, uncle, and sister Amruta for their encouragement and unrelenting support in completing the doctoral thesis journey and for giving there everything so that I can fulfill my research endeavors in Europe. Words are not enough to describe how grateful I am to them.

Experiments presented in this thesis were carried out using the Grid5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER, and several Universities as well as other organizations. (see https ://www.grid5000.fr)

Dedicated to my uncle, Mr. Anil Kulkarni,
for continual encouragement and wisdom

Contents

List of Figures

List of Tables

List of Abbreviations

bap ..... band aperiodicity
BAP ..... Band aperiodicity distortion
CNN ..... Convolutional Neural Network
CRNN ..... Convolutional Recurrent Neural Network
DNN ..... Deep Neural Network
DPM ..... Diffusion Probabilistic Model
GAN ..... Generative Adversarial Network
Glow ..... Generative Flow
GMVAE ..... Gaussian Mixture Variational Autoencoder
GRU ..... Gated Recurrent Unit
GST ..... Global Style Token
HMM ..... Hidden Markov Model
IAF ..... Inverse Autoregressive Flow
Lf0 ..... Log fundamental frequency
LSTM ..... Long Short Term Memory
MCD ..... Mel Cepstral Distortion
MFCC ..... Mel Frequency Cepstral Coefficient
MGC ..... Mel Generalized Cepstrum
MOS ..... Mean Opinion Score
MSE ..... Mean Squared Error
RCVAE ..... Recurrent Conditional Variational Autoencoder
RMSE ..... Root Mean Squared Error
RNN ..... Recurrent Neural Network
TTS ..... Text To Speech
VAE ..... Variational Autoencoder
vuv ..... voiced unvoiced

Chapter 1

Introduction

"The study of expression is difficult, owing to the movements being often extremely slight, and of a fleeting nature. A difference may be clearly percieved, and yet it may be impossible."

— *Charles Darwin, The expression of the emotions in man and animals, 1872*

1.1 Motivation

Speech is one of the fascinating means of human communication, and the ability to express inner feelings and thoughts is another. With the evolution of technology, the idea of machines, computers, and robots being able to talk and express themselves as human beings has become a reality with speech synthesis. Speech synthesis produces artificial speech generated by a computer/machine from input text sentences. The main goal of text-to-speech (TTS) synthesis is to produce speech sounds of quality comparable to human speech. A sentence spoken by a person innately possesses verbal communication through expressivity as an intrinsic feature delivered subconsciously. Expressive speech synthesis aims at generating synthesized speech that exhibits expressivity such as emotion, speaking style, etc., reflecting the complex emotional states from a textual sentence. A speech synthesis system should assign certain linguistic factors such as intonation, rhythm, stress to make the artificially produced speech more realistic.

The term expressivity in speech usually refers to the speech characteristics featuring emotions, speaking style, the relationship of speech with gestures, and facial expression. Throughout this work, we focused only on the emotional aspect of expressivity in speech without considering the textual content in speech. Every sentence spoken by a human inherently comprises expressiveness, conditioned on the pragmatics behind the content of the speech. The term expressivity has been used to describe how objects can stimulate a feeling or emotion [Hook et al., 2011], as well as the degree to which a person expresses emotions [Ståhl et al., 2005, Wensveen et al., 2000]. In [Dalsgård and Hansen, 2008, Wensveen et al., 2002, Dalsgaard et al., 2016], it is illustrated that expressiveness of interactions with the human-machine interface through multimodal expressions such as text, audio or video depicts the correlation with user's emotional state of mind. Therefore,

future interactive systems need to take into account emotional attributes to be truly adaptive [Wensveen et al., 2000, Bruns et al., 2021].

One of many classical applications of TTS includes smart-home devices, announcement systems, computer-assisted learning, virtual assistant systems, visually-impaired assistance, a book reader applications. The lack of expressivity in human-machine interactions may lead to poor user experience due to a high degree of monotony in synthesized speech [Kim et al., 2004]. The research in expressive TTS has been revamped after the commercial success of deep learning techniques in parametric TTS frameworks and end-to-end TTS. Despite significant advances in the overall quality of speech synthesis, existing techniques lack the prospect of expressivity as an inherent aspect of human verbal communication. The next challenging aspect in the speech synthesis domain is to enable expressivity as an intrinsic factor in synthesizing speech.

Furthermore, the reading style or expressions recorded for a specific speaker's voice only allow the system to synthesize in that particular reading style or expression. Data availability during the training phase significantly impacts the quality, naturalness, and intelligibility. As a corollary, to produce expressive voice synthesis for a new speaker, we must first create a speech dataset with various emotions. Recording an expressive speech dataset to construct an expressive speech synthesis system for every new speaker's voice is inconvenient.

Developing an expressive speech dataset is time-consuming and costly in terms of data collection, recording, labeling, alignment, and evaluation workload. Moreover, the speaker responsible for providing expressive speech must enact the emotions while recording. Many methodologies leverage audiobooks to extract expressive speech data to avoid the speech acquisition process [Charfuelan and Steiner, 2013]. Due to the various possible variations in a single emotion, labeling expressions is a difficult task. This creates a significant hurdle in the advancement of expressive speech synthesis.

This thesis work proposes deep learning architectures to transfer the expressivity from an existing expressive speech dataset to a target speaker's neutral TTS system. Hence, the thesis aims to eliminate the need to explicitly create an expressive speech dataset for speakers for which only a neutral voice is available. The main challenge in expressivity transfer is to disentangle speaker characteristics and expressiveness and transfer it to the desired speaker voice. Apart from creating expressive TTS in the target speaker's voice, the same technique can be deployed in assisting voice dubbing applications as well as the gaming industry to improve the efficiency in controlling the expressivity in various scenarios [Akhulkova et al., 2021]. The dubbing studios and theater directors would be interested in transforming and synthesizing expressive voices for automatic switch between different speaker voices and expressivity classes [Beller et al., 2006, Beller, 2010].

Recent research work has shown state-of-the-art results for speaker adaptation [Casanova et al., 2021, Jia et al., 2018], voice conversion [Zhang et al., 2021], and stylistic transfer in image processing [Jing et al., 2020], these studies rely on transfer learning in the context of deep learning techniques. For example, texture transfer from the style of one image into another image is performed under the constraint of preserving semantic content of target image [Gatys et al., 2016]. Contrary to image style transfer, the main challenge in the context of expressivity transfer in the TTS system is to derive temporal variations across the speech waveforms conditioned on the textual content for a well-defined ex-

pressivity class. We propose approaches to create latent expressivity representation and usage of learned latent representation to transfer expressivity to new speaker's voices in the synthesis phase.

1.2 Expressivity

In the 1960s, Paul Ekman visited New Guinea, Borneo, Brazil, Japan, and the United States to research the effect of facial emotional expressivity in a pan-cultural environment [Ekman et al., 1969]. His research work supported Darwin's proposition [Darwin, 1872] that facial expressions of emotion are similar among humans, regardless of culture, because of their evolutionary origin. Over the years, expressivity psychological research has been conducted in various modalities such as facial expressions [Kinchella and Guo, 2021], body movements [Dael et al., 2012], expressivity in musical performances [Woody, 2000], and non-verbal expressions [Aranguren and Tonnelat, 2014].

Expressivity term is derived from the word expressive. The dictionary meaning of expressive is "effectively conveys thought or feeling", which comes from a Latine word experiment meaning "to press out". The emotional aspect of expressivity is an integral part of human speech, without which it is impossible to convey one's inner feelings and thoughts. Emotional expressiveness has been the center of interest to natural scientists and psychologists since the time of Charles Darwin and William James [Riggio and Riggio, 2002]. The non-verbal expression of emotions remains a topic of great interest to researchers in interpersonal communication. Emotions have long been associated with nonverbal facial expressions and expressive gestures. The person's ability to accurately send and receive such nonverbal information can be a significant factor affecting their ability to communicate with each other [Buck et al., 1972].

Understanding meaningful emotional expressiveness prevails as a challenge for researchers to perceive the fundamental essence of expressiveness. Expressiveness helps in encoding the inner feelings with non-verbal expressions as well as from one's vocal tone. It is theorized that, in general, the ability to express emotions, especially the ability to encode emotions, may represent a central component of an individual's personality because emotional communication plays a vital role in interacting as well as developing relationships [Friedman, 1979]. Moreover, as a personal style, emotional expression is relatively consistent in different situations and throughout the development process [Allport and Vernon, 1933].

The most widely used methodologies for studying emotional expressivity in speech include Ekman's Basic Emotion Theory and the Russell circumplex model. The circumplex model proposed by Russell is based on a subjective study of emotional words. James Russell developed a circumplex model for creating a circular representation of emotions based on two independent and bipolar factors, namely valence and activation [Russell, 1980]. The valence refers to the degree of pleasantness and unpleasantness, while activation describes the high or low arousal. The circumplex emotion model includes valence and activation as horizontal and vertical dimensions in defining circular variations across the emotion states, as shown in Figure 1.1. Hence, defining circular emotional models enables us to understand variability in mixed emotions across quadrants of the circumplex

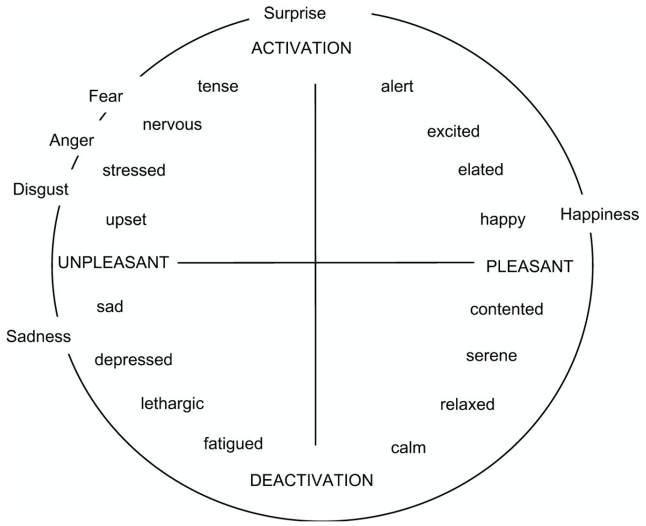

Figure 1.1 – Russells circumplex model for emotions

models.

Furthermore, the circumplex model asserts that opposite emotions such as surprise and calm cannot be classified as mixed emotions because they have different levels of valence or arousal.

Interestingly, Ekman proposed a theory of basic emotions to juxtapose Russell's circumplex model to define distinctive discrete states of emotions such as anger, fear, joy, sadness, and disgust. Ekman published his work on the basic theory of emotions based on a study conducted on facial expressions in deliberate and spontaneous environments [Ekman, 1992]. Each emotional state has characteristics in common with other emotions: rapid onset, short duration, unbidden occurrence, automatic appraisal, and coherence among responses [Ekman, 1992]. In the scope of this thesis study, we use audio aspects of the expressive audio-visual dataset published in [Dahmani et al., 2019] to undertake expressivity transfer. The expressive audio-visual dataset is designed in accordance with Ekamn's basic theory of emotions [Dalgleish and Power, 1999].

1.3 Thesis Objective

The main objective of the thesis work is to develop an expressive TTS system in a speaker's voice for which only neutral speech data is available. To achieve this goal, knowledge of expressivity attributes is extracted from an existing expressive speech dataset. Afterward, expressivity knowledge is transferred to the target speaker's voice in the TTS framework. Therefore, this enables the synthesis of expressive speech in the target speaker's voice without explicitly acquiring the expressive speech data in the target speaker's voice. The synthesized speech is primarily determined by the speech data used in training the TTS system. From Figure 1.2, if TTS A is trained using the neutral speech data of speaker A, the TTS A system will only be able to synthesize speech in speaker A's neutral voice. In the expressive TTS system depicted in Figure 1.2 as TTS

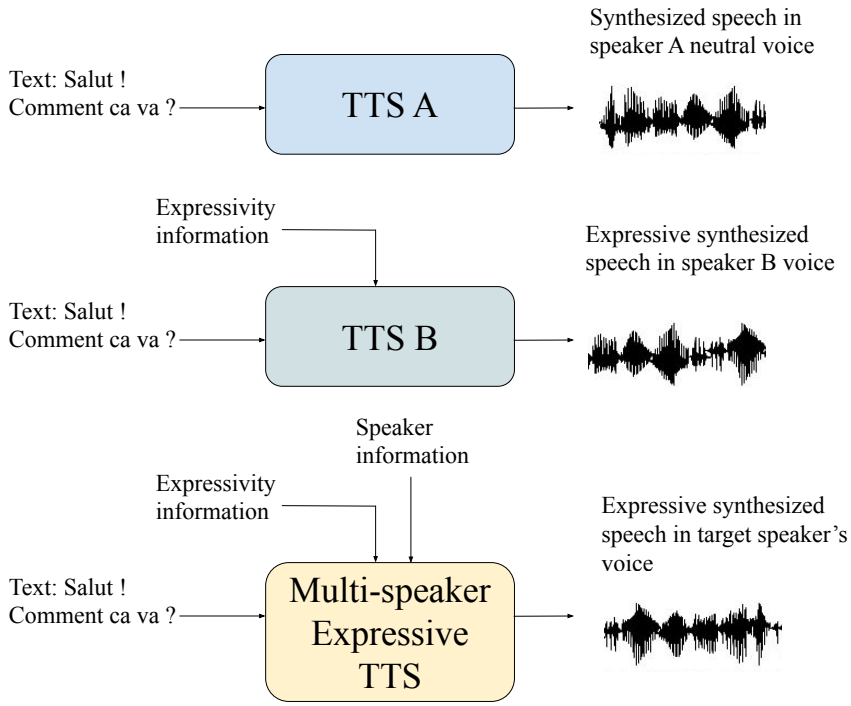

Figure 1.2 – General approches towards developing TTS systems, where TTS A is in speaker A neutral voice, TTS B is expressive TTS in speaker B's voice. Multi-speaker expressive TTS is in both speaker A and speaker B voices, able to synthesize expressive speech even for speaker A

B, in which explicit expressivity class information is provided as additional input along with text to the TTS B system. The addition of expressivity class in TTS B enables to synthesize expressive speech in speaker B. For expressivity transfer, the goal is to transfer expressivity to speaker A's voice in a multispeaker setting. Therefore, the thesis work explores deep learning approaches for creating an expressive TTS system in speaker A's voice.

The thesis work investigates the transfer of expressivity approaches in a multispeaker expressive TTS setting (bottom part of Figure 1.2). The proposed approaches are designed in a multispeaker TTS system for which required inputs include speaker information, expressivity class information, and text. The presented work focuses on creating latent representations of speaker and expressivity. These representations are then used to interpolate in the expressivity transfer phase. The challenging aspect of developing latent representation is to disentangle the speaker's information and expressivity. One of the main challenges of this task is to transfer the expressive attributes from available expressive speech data and ensure that the target speaker's voice is unaltered during the transfer of expressivity. The focus of the work is on investigating transfer learning mechanisms, which will accelerate the efforts toward exploiting existing expressive speech datasets for the French language. The scope of work incorporates expressivity as an emotion attribute with well-defined emotion classes described as expressivity classes throughout the thesis.

1.4 Thesis Contributions

As discussed earlier, we investigated the expressivity transfer mechanism developed using multispeaker expressive TTS for French. The thesis investigates three major frameworks for developing multispeaker expressive TTS systems, namely parametric TTS, end-to-end autoregressive TTS, and end-to-end non-autoregressive TTS. We developed proposed TTS systems using deep neural network architectures. The main contributions of the thesis work are as given below,

1. We proposed to use a metric learning loss in the training of variational autoencoder based acoustic model for developing expressive multispeaker parametric TTS system. The metric learning loss function is used for enforcing better clustering of expressivity in latent space representation. We extended this approach to an inverse autoregressive Flow (IAF) architecture for replacing variational inference performed using Kullback–Leibler (KL) divergence. We proposed novel Flow metric learning as augmentation of metric learning loss with IAF architecture. To our knowledge, the presented Flow metric learning is the first framework proposed in the context of deep learning and for enhancing expressivity representation in the TTS system.
2. Similar to the parametric TTS framework, we presented an extension to the existing autoregressive end-to-end TTS system by jointly training with a metric learning loss function. After that, we presented a multistage attention architecture for fine-grained expressivity transfer in an end-to-end TTS framework. Fine-grained expressivity transfer enables the construction of text-dependent expressivity representations at the local level and fixed dimensional expressivity representation at the global level.
3. We presented a novel expressivity encoder implemented with self-attention statistical pooling to create fixed dimensional expressivity representation, which extracts the statistics by focusing on vital expressivity information from frames level representation. Furthermore, we proposed a multiscale expressivity encoder that incorporates acoustic features used in classical emotion recognition systems.
4. We proposed expressivity encoder extension to non-autoregressive TTS systems based on deep generative models, Generative Flow (Glow), and Diffusion probabilistic models (DPM). The usage of Glow and DPM provides exact log-likelihood estimates, thus providing high-quality synthesis, flexible sampling, and inference speed.

In addition to the above, we present a transfer learning approach to create latent speaker representation using a pretrained speaker recognition model to enable multispeaker functionality in the parametric TTS framework. Also, the evaluation of expressivity transfer is a challenging aspect of the thesis, as no reference expressive speech data is available in the target speaker's voice. Therefore, we proposed two subjective evaluation metrics, speaker mean opinion score (MOS) and expressive MOS, which signifies the performance measure of framework to transfer the expressivity and the retention of the target speaker's voice. We propose a cosine similarity as an objective evaluation metric

to measure the strength of expressivity and the speaker's voice.

Parts of the thesis have been published in the following articles:

1. **Ajinkya Kulkarni**, Vincent Colotte, Denis Jouvet, Improving transfer of expressivity for end-to-end multispeaker text-to-speech synthesis, In *29th European Signal Processing Conference*, 2021, Ireland.
2. **Ajinkya Kulkarni**, Vincent Colotte, Denis Jouvet, Transfer learning of the expressivity using Flow metric learning in multispeaker text-to-speech synthesis, In *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2020, China.
3. **Ajinkya Kulkarni**, Vincent Colotte, Denis Jouvet, Deep variational metric learning for transfer of expressivity in multispeaker text to speech, In *Proceedings of the 8th International Conference on Statistical Language and Speech Processing*, 2020 United Kingdom.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows.

1. Chapter 2. discusses various aspects of state-of-the-art text-to-speech systems used in the scope of thesis work, which includes deep learning techniques, speech synthesis systems, vocoders, and expressivity transfer approaches.
2. Chapter 3 contains information about the speech datasets used in the thesis work. We also go into the specifics of text preprocessing and acoustic feature processing techniques utilized in the context of the thesis' TTS systems.
3. Chapter 4 discusses evaluation metrics for measuring the performance of text-to-speech systems and expressivity transfer. It explains the objective measures used to evaluate the prediction of acoustic features. Following that, it elaborates on listening tests used to determine subjective evaluation scores.
4. Chapter 5 describes the proposed methodologies for expressivity transfer developed under the parametric TTS framework. It addresses single speaker parametric TTS systems initially, followed by proposed multispeaker expressive parametric TTS systems for expressivity transfer.
5. Chapter 6 describes the proposed methodologies for expressivity transfer established within the end-to-end TTS framework. It focuses on the addition of expressivity in autoregressive and non-autoregressive TTS frameworks.
6. Chapter 7 summarizes the thesis and provides future research directions in the context of expressivity transfer in TTS.

Chapter 2

Text to Speech : State of the art

2.1 Introduction

Speech and language have emerged as the primary means of communication throughout the history of mankind. Christian Gottlieb Kratzenstein, the German-Danish scholar in 1791, devised the human vocal tract model for vowel sounds [Flanagan, 1971, Flanagan and Rabiner, 1973], which was the first attempt to develop a speech synthesis system. He built acoustic resonators inspired by the human vocal tract and triggered the response from resonators for producing five long vowel sounds, namely /a/, /e/, /i/, /o/, and /u/. In 1928 Bell Labs researcher Dudley proposed transmission of spectrum information rather than speech signal, which led to the invention of vocoder [Schroeder, 1999]. This apparatus was demonstrated at New York World Fair in 1939 as the first electric speaking machine. This innovation was incorporated as means of encrypted communications during World War II between Allied military headquarters across five continents. The principle idea behind vocoder is still extensively assimilated in modern speech synthesis systems to generate the speech waveform from acoustic features. The scientific progress in electronic devices and ability to process large computations facilitated the deeper understanding of languages and human speech production, which requires the expertise in various domains such as linguistics [Flanagan and Rabiner, 1973], acoustics [Kinsler et al., 1999], digital signal processing [William D Stanley and Saunders, 1988], machine learning [Tan et al., 2021, Bishop, 2006, Jordan and Mitchell, 2015].

George Moore postulated that approximately every two years, the number of transistors on microchips would double, which is commonly referred to as Moore's Law [Moore, 1998]. This idea was first proposed in 1965, and Moore's Law has been supported by computing development over the past few decades. The overall growth in technological aspects of computations empowered the machines to train neural network models with more layers and a large amount of data referred to as deep neural networks. The speech synthesis domain is no exception to these deep learning advancements and has seen remarkable improvements in synthesized speech.

To begin with, we introduce the basic notions of deep learning and relevant architectures of deep neural networks in Section 2.2. Furthermore, we provide a detailed explanation of generative models and deep generative models in Section 2.3, used in the-

sis work for various tasks, such as vocoder, TTS systems, expressivity encoder network, etc. Section 2.4 explains deep metric learning and its usage in discriminative tasks. We introduce basic aspects of speech in Section 2.5. After that, we will explain the basic building blocks of Text-to-Speech (TTS) in Section 2.6. Section 2.7 describes the vocoder and various approaches for generating speech waveforms from acoustic features. Section 2.8 describes the state-of-art end-to-end TTS systems based on deep neural networks. After that, Section 2.9 details various methodologies to create expressive TTS systems and transfer of expressivity is explained in Section 2.10.

2.2 Deep Learning

Before going in-depth into deep learning, we will first discuss the basic notions of artificial neural networks (ANN) in the context of supervised machine learning. Artificial neural networks were developed as mathematical formulation providing connectionist approach for reasoning the information similar to capabilities of biological neurons [Rosenblatt, 1958, Mullin and Rosenblatt, 1962, McCulloch and Pitts, 1990]. The idea of activations and connected nodes states the electrical activity in brain neurons, modeling the biological model of neuron activation. The basic element in ANN is a single-cell perceptron, which computes the weighted sum of input units transformed through a non-linear activation function. One of the most popular ANNs is multilayer perceptron, often called feedforward neural networks. The feedforward networks are acyclic in nature, i.e., no information is passed through time. The feedforward networks consist of input, hidden, and output layers implemented as stacked nodes connected through each layer. Therefore, the feedforward network can perform a non-linear transformation to extract the underlying complex arbitrary distribution present in provided input.

The most commonly used activation functions were hyperbolic tangent and sigmoid, defined in Equation 2.1, Equation 2.2, respectively, where input, x is given to activation function θ to produce non-linear deterministic output. Various activation functions have been proposed over the years, such as Rectified Linear Unit (ReLU), Exponential Linear Unit (ELU), Scaled Exponential Linear Unit (SeLU), Gaussian Error Linear Unit (GeLU) [Apicella et al., 2021, Klambauer et al., 2017, Hendrycks and Gimpel, 2016] showing improvement over standard machine learning datasets.

$$\theta_{tanh} = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{2.1}$$

$$\theta_{sigmoid} = \frac{1}{1 + e^{-x}} \tag{2.2}$$

In Equation 2.3 and Equation 2.4, the forward pass for a feedforward neural network is defined as a matrix multiplied with the neural network's weights for a given input x_i of dimension I. In addition, the output of Equation 2.3, a_h is fed into the activation function θ_{actfn}, which performs the non-linear transformation of input data as output, $\tilde{x}_h$.

$$a_h = \sum_{i=1}^{I} w_{i,h} x_i \tag{2.3}$$

$$\tilde{x}_h = \theta_{actfn}(a_h) \tag{2.4}$$

The biological processes inspired convolutional neural networks (CNN) [Hubel and Wiesel, 1968, Fukushima, 1980, Matsugu et al., 2003] explore the pattern of connectivity between neurons resembles like organization of animal visual cortex. Inspired by the work done in [Fukushima, 1980] first application of convolutional neural networks was presented in [LeCun et al., 1989] illustrating learnable feature extraction on visual data. The CNN comprises convolution operations using filters to extract the feature maps from input, thus exploiting the spatial information. Then, a pooling operation is applied for reducing the dimensionality of feature maps [Zhu et al., 2018a].

Recurrent neural network (RNN) is another type of ANN, which was based on work presented in [Rumelhart et al., 1987], and Hopfield network [Hopfield, 1982]. The RNNs are directed graphs of connected nodes for processing temporal sequence, which has cyclic connections to access the temporal sequence. Therefore, RNNs illustrate the sequential non-linear transformation by taking contextual information from previously provided input states into account, to generate output representation $\tilde{x}_h^t$ for the current time sequence, t. Lets consider input data, x of length T given to RNN of I input units, and H hidden units, then output of RNN is estimated from previously predicted output, $\tilde{x}_h^{t-1}$ and current input x^t as given below,

$$a_h^t = \sum_{i=1}^{I} w_{i,h} x_i^t + \sum_{h'=1}^{H} w_{h',h} \tilde{x}_{h'}^{t-1} \tag{2.5}$$

$$\tilde{x}_h^t = \theta_{actfn}(a_h^t) \tag{2.6}$$

The classical RNNs have a finite memory state to process the sequential information. The RNNs are principally difficult to train due to long-range temporal dependencies leading to exploding or vanishing gradient problems [Hochreiter, 1998]. Therefore, the need to access long-temporal context leads the network to output decay or alleviate exponential around the cyclic connections. In 1997, [Hochreiter and Schmidhuber, 1997] proposed Long-short-term-memory (LSTM) as a way to realize a memory cell for controlling the flow of context information using gating and storage mechanism. The proposed approach was further refined using peephole connection [Gers and Schmidhuber, 2000], and later using forget gate [Graves et al., 2013]. Few variants of RNN's are proposed that accesses context from both directions of sequences, such architecture in combination with LSTM, called Bidirectional LSTM (BLSTM). Another variant of the memory cell-based architecture for RNNs was presented as a Gated recurrent unit (GRU), having fewer parameters than LSTM without having an output gate [Cho et al., 2014, Gers et al., 2000]. The progress in RNNs over the 21st century revamped the sequence to sequence learning applications [Cho et al., 2014], and now it is one of the most commonly used frameworks in deep learning architectures.

In a supervised learning setting, the most crucial stage is to train ANN architecture on input-output pairs. Before initiating the training phase, weights of ANNs are initialized randomly or by various techniques for weight initializations [Sousa, 2016, Boulila et al., 2021, Glorot and Bengio, 2010]. The training phase consists of a forward pass where

input is passed through the ANNs and output is estimated. Then, an appropriate task-dependent loss function is used to compute the error gradient to measure how close the predicted output is to the actual output. The most widely used technique for computing the error gradient is the backpropagation algorithm introduced in 1986 [Rumelhart et al., 1986]. The backpropagation algorithm uses linearization and dynamic programming to compute the derivatives [Goodfellow et al., 2016]. The gradients of the loss function with respect to weights in each layer are estimated using the chain rule. After computing the error gradients, the optimization strategy focuses on how the weight is updated to reduce the loss computed in the forward pass and update the weights in the most precise manner. The most commonly used optimization strategies are gradient descent [LeCun et al., 1998], stochastic gradient descent [Lin et al., 2020], and Adam optimizer [Kingma and Ba, 2015]. The error gradients are computed with respect to loss functions used to measure the error value as a comparison between predicted output and desired output. The most commonly used loss function for regression tasks is mean squared error (MSE). The loss term is computed as the squared difference between desired output, y and predicted output, $\hat{y}$ as described by Equation 2.7, where N is a number of samples. If the distribution of output data conditioned on input is normally distributed around a mean value, MSE shows better performance.

$$\mathcal{L}oss_{MSE} = \frac{1}{N}\sum_{i=0}^{N}(y_i - \hat{y_i})^2 \tag{2.7}$$

The cross-entropy loss is used for discriminative tasks such as image classification, and speaker recognition in a multiclass setting. The cross-entropy loss function is based on entropy and information theory concepts. Entropy is the required number of bits to transmit a stochastic event selected from a probability distribution [Murphy, 2012]. The cross-entropy loss measures the difference between the probability distribution of predicted output classes and true output classes. Therefore, the cross-entropy loss, $\mathcal{L}oss_{cross-entropy}$ can be defined as Equation 2.8 for true output y, and predicted output $\hat{y}$ of N examples, and K classes. For more details regarding the basics of ANNs reader is suggested to refer to [Bishop and Nasrabadi, 2007].

$$\mathcal{L}oss_{cross-entropy} = -\sum_{i=0}^{N}\sum_{k=0}^{K} y_{i,k} log(\hat{y}_{i,k}) \tag{2.8}$$

where $y_{i,k}$ is the binary target category. And $\hat{y_{i,k}}$ is the output of the model for item i, and class k, it is the predicted probability of class k for item i.

With the advancement in computation resources over the last decades, researchers trained ANNs with many hidden units with an increasing number of layers replacing shallow architectures. Therefore, the term "deep" in the context of deep learning refers to these deeper connections across many hidden layers. In [Bengio, 2012], Yoshua Bengio stated, "Deep learning algorithms seek to exploit the unknown structure in the input distribution to discover good representations, often at multiple levels, with higher-level learned features defined in terms of lower-level features." The growth in computation power and data availability in recent years enabled the capability of deep learning methods. Geoffrey Hinton outlined the importance of computation power by stating, "It has

been obvious since the 1980s that backpropagation through deep autoencoders would be very effective for nonlinear dimensionality reduction, provided that computers were fast enough, data sets were big enough, and the initial weights were close enough to a good solution. All three conditions are now satisfied" [Hinton and Salakhutdinov, 2006]. Advancements in neural networks lead to various techniques for designing and training end-to-end deep neural network systems with backpropagation algorithm. New mechanisms have been proposed auxiliary neural network modules to equip deep architectures as single trainable systems like attention mechanism [Chaudhari et al., 2019], and normalization techniques namely layer normalization [Ba et al., 2016], batch normalization [Ioffe and Szegedy, 2015], activation normalization [Kingma and Dhariwal, 2018]. The book titled "Deep learning" provides concepts used in the modern deep learning field; for more detailed descriptions, the reader is recommended to refer to this book [Goodfellow et al., 2016].

2.3 Generative Models

Generative models developed under deep learning framework are trained to approximate complex, high-dimensional probablity distributions using large number of samples [Ruthotto and Haber, 2021]. In 1983 Geoffrey Hinton first presented idea of Boltzman machines as generative model to tackle unsupervised clustering task [Hinton and Sejnowski, 1983]. Recently, many frameworks of generative models are presented for solving data synthesis [Park et al., 2018], clustering [Broad et al., 2021, Yang et al., 2020b], denoising [Ashfahani et al., 2020], style transfer [Kingma and Dhariwal, 2018, Wilson et al., 2021] in speech, image and text modalities [Toshevska and Gievska, 2021, Luan et al., 2017, Zhou et al., 2021]. Most studies of generative models includes models such as PixelRNN [van den Oord et al., 2016], variational autoencoders [Kingma and Welling, 2014], generative adversarial networks [Goodfellow et al., 2014], real valued non-volume preserving transformations [Dinh et al., 2017], Generative Flow [Kingma and Dhariwal, 2018], FFJORD [Grathwohl et al., 2019], and diffusion probablistic models [Sohl-Dickstein et al., 2015]. As only finite number of samples are available in practice, performance of generative models heavily relies on choices of neural network architecture, loss criterion, regularization, and training procedure.

The core principle behind the generative model is to learn the true probabilistic distribution of data, $p(x)$, where x is of $\mathbb{R}^n$ of n dimension, and transformation function to known simple distribution such as Gaussian distribution to a latent space of known dimension. Such transformation to simple tractable distribution gives access to interpretation and manipulation of latent attributes of data to represented as latent space distribution, $p(z)$, where z is of $\mathbb{R}^m$ dimension. Therefore, for $x \sim p(x)$ there exists at least one point in latent space, $z \sim p(z)$. The transformation function for a generative model is denoted as g_θ, where θ represents parameters of neural network.

$$x \sim g_\theta(z) \tag{2.9}$$

$$p(x) = \int p(x|z)p(z)\,dz \tag{2.10}$$

The main functionality of g_θ is to map point z from $p(z)$ to point x of output distribution $p(x)$, as Equation 2.9. Furthermore, generator g_θ is also used to estimate the likelihood or evidence of given point x from output distribution $p(x)$, under assumption that generator g_θ is known as stated in Equation 2.10. The likelihood $p(x|z)$ reflects how close $g_\theta(z)$ is to x. As stated earlier most commonly used latent distribution is Gaussian distribution. Therefore, the likelihood estimate in Equation 2.10 can be rewritten as Equation 2.11, where variance σ controls the likelihood probability around the samples.

$$p(x|z) = (2\pi\sigma^2)^{-\frac{n}{2}} e^{\left(-\frac{1}{2\sigma^2}\|g_\theta(z)-x\|^2\right)} \tag{2.11}$$

Generative models have been used either as a generator network to predict acoustic features or create embedding representations for expressivity. The generator network for acoustic features is usually trained by conditioning latent distribution with information from various sources such as text, speaker, and expressivity. On the other hand, expressivity representation is learned through generative models that are used to disentangle the latent attributes represented by simple Gaussian distribution. We will overview generative models such as variational autoencoder (VAE), and inverse autoregressive Flow (IAF) used in the context of creating a meaningful representation of expressivity. Thereafter we explain Generative Flow (Glow) and diffusion probabilistic models (DPM), which are used as acoustic feature generators for developing expressive end-to-end TTS systems.

2.3.1 Variational Autoencoders

Autoencoder is a neural network designed to learn an identity function in an unsupervised way to reconstruct the original input while compressing the data in the process to discover a more efficient and compressed representation. The idea was originated in the 1980s and later promoted by the seminal paper by Hinton Salakhutdinov [Weng, 2018,Hinton and Salakhutdinov, 2006]. The principle of Variational Autoencoder [Kingma and Welling, 2014], in short VAE, is less similar to any of the autoencoder models. Still, it is heavily anchored in variational bayesian and graphical model approaches. In the context of expressivity transfer, the TTS system needs to map the expressivity representation to a predefined probability distribution. This assists in creating a disentangled representation of expressivity onto latent space representation and can be used to synthesize desired expressivity in the synthesis phase irrespective of the speaker's voice.

Variational Autoencoder

Variational autoencoders were introduced in 2013 by Kingma [Kingma and Welling, 2014], and Rezende [Rezende et al., 2014] independently. Variational autoencoders have similar components as autoencoders, that is an encoder, a decoder, and a loss function as illustrated in Figure 2.1. However, for training, the loss function corresponds to the reconstruction error (as for the autoencoders) plus a regularization term defined with a Kullback-Leibler (KL) divergence. The encoder takes an input data x and represents it as

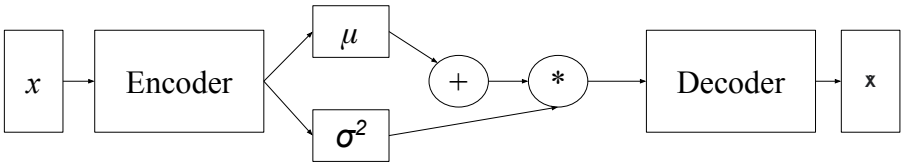

Figure 2.1 – Schematics of a training-time variational autoencoder with reparameterization trick

latent representation z. Thus, the encoder learns to compress the input data into a lower dimension space. Let's denote $Q_\Theta(z|x)$ the encoder distribution, which is a Gaussian probability density with parameters Θ. The decoder takes the latent representation z as input and output reconstructed data sampled from the Gaussian distribution. The distribution of the decoder output is denoted by $P_\phi(x|z)$ with parameters ϕ.

$$Loss(\Theta, \phi) = E_{z \sim Q_\Theta(z|x)}[logP_\phi(x|z)] + KL(Q_\Theta(z|x)||P(z)) \tag{2.12}$$

From Equation 2.12, the loss function of variational autoencoder has two terms, the first term is a reconstruction loss or expected log-likelihood, which represents the expectation over the reconstruction of input. This term indicates how well the decoder learns to reconstruct the input data. The second term is a regularizer which is the KL divergence between the encoder's distribution $Q_\Theta(z|x)$ and true prior distribution $P(z)$ as mentioned in Equation 2.12. This measure indicates how close the learned distribution $Q_\Theta(z|x)$ is to the true prior distribution $P(z)$. The true prior distribution, $P(z)$ is specified as a normal distribution with zero mean and unit variance in the variational autoencoder. Thus, the KL divergence term will approximate the latent representation z to have a normal distribution to avoid the penalty. Without the regularizer penalty, the encoder will map the latent state for each data point in different parts of the Euclidean space. As we want the latent space representation meaningful, we penalize it with a KL divergence to keep the similar representation of data points close to each other. During the inference, we sample from latent space, and the decoder network acts as a generative model with the ability to generate new data points similar to the examples seen during the training.

For the implementation of variational autoencoders, the encoder model output two parameters, that is, the mean and the variance to describe the latent space distribution. During the training phase of variational autoencoders, we estimate the error gradient with respect to the final output and backpropagate the error and update the parameters using a stochastic gradient descent algorithm. As a stochastic gradient can be applied to stochastic inputs but not stochastic units within a network, to resolve this, we use the reparameterization trick [Kingma and Welling, 2014]. In reparameterization, we randomly sample ϵ from the normal distribution and then shift sampled ϵ randomly with latent distribution's mean μ and then scaling it with the variance σ^2 of latent distribution, as shown in Equation 2.13.

$$z = \mu + \epsilon \cdot \sigma \tag{2.13}$$

Conditional Variational Autoencoder

As mentioned earlier, the decoder of a variational autoencoder acts as a generative model, which takes as input a latent variable sampled from the normal distribution. However, with this model, we have no direct control over the generation process. In a traditional variational autoencoder, information about the type of input data is not taken into consideration. Equation 2.14 shows a modification of the loss function that conditions the encoder and decoder distribution on a conditional variable c, which provides information about the type of input data that is mapped into the latent space. Therefore, the resulting distributions of the encoder and of the decoder conditioned on c is given by $Q_\Theta(z|x,c)$ and $P_\phi(x|z,c)$ and true prior distribution as $P(z|c)$. These are conditional probability distributions, which show that for each possible condition c, there exists prior distribution $P(z|c)$. Conditional variational autoencoders allow us to handle input to output mapping as one to many learning problems without explicitly defining the output distribution [Rezende et al., 2014].

$$Loss(\Theta,\phi) = E_{z\sim Q_\Theta(z|x,c)}[logP_\phi(x|z,c)] + KL(Q_\Theta(z|x,c)||P(z|c)) \tag{2.14}$$

2.3.2 Normalizing Flow

The promising results shown by variational inference in Bayesian learning created state-of-art results in various applications including semi-supervised classification [Kingma et al., 2014], generative models of images [El-Kaddoury et al., 2019], voice conversion [van den Oord et al., 2017]. Variational inference is widely used as a tool for investigating latent space for analysis of semantic information [Bengio et al., 2013]. Despite these state-of-the-art results, the ability of variational inference is constrained due to intractable posterior distributions to be approximated by the class of known probability distributions, over which we search for the best approximation to the true posterior distribution [Rezende and Mohamed, 2015]. The central issue in variational inference is the selection of approximate posterior distribution.

Normalizing Flow as generative models were first described in non-linear independent components estimation (NICE) [Dinh et al., 2015] and extended in real non-volume preserving (Real NVP) [Dinh et al., 2017]. Normalizing Flow model's posterior distributions constructed using series of cascaded invertible transformations to map simple initial distribution to arbitrary complex, flexible distribution with tractable Jacobian [Kingma et al., 2016]. These cascaded invertible transformations are called normalizing Flow.

Lets consider x as random vector with unknown true distribution $x \sim P(x)$, and we define this true distribution with model parameter θ as $P_\theta(x)$. Thereafter generative model, g_θ can describe generative process as stated Equation 2.9, where z is latent variable sampled from prior distribution $P_\phi(z)$, which has simple tractable density such as multivariate Gaussian distribution, $P_\phi(z) = \mathcal{N}(z;0,I)$. In normalizing Flow, function g_θ is designed to be bijective function as described by Equation 2.15. Therefore, by definition of probability distribution we can rewrite Equation 2.16 as Equation 2.17 according to change of variable theorem.

$$z = g_\theta^{-1}(x) = f_\theta(x) \tag{2.15}$$

$$\int P_\theta(x)\, dx = \int P_\phi(z)\, dz = 1 \tag{2.16}$$

$$logP_\theta(x) = logP_\phi(z) + log|det\left(\frac{dz}{dx}\right)| \tag{2.17}$$

Above Equation 2.17 estimates change in density when data point x is transformed to latent variable z using Equation 2.15. The name normalizing Flow can be constituted as normalizing means that the change of variables gives as normalized densities after the application of invertible transformation. The Flow term illustrates that invertible transformation can be cascaded with each other to create more complex invertible transformation. After K invertible transformation, Equation 2.17 can be stated as,

$$logP_\theta(x) = logP_\phi(z) + \sum_{k=1,z_0=x}^{k=K,z_K=z} log|det\left(\frac{dz_k}{dz_{k-1}}\right)| \tag{2.18}$$

The second term in Equation 2.18 represents the change in log densities going from z_{k-1} to z_k under transformations $f_{\theta k}$ for K steps of normalizing Flows. Furthermore, the design of normalizing Flow needs to satisfy two conditions, firstly, transformations must be invertible. And second, easy computation of Jacobian determinant. The loss criterion for model g_θ for defining data distribution $P_\theta(x)$ with model parameter θ over dataset $\mathcal{D}$ of data point x can be written as Equation 2.19 for estimating negative log-likelihood as given below,

$$\mathcal{L}(\mathcal{D}) = -\frac{1}{N}\sum logP_\theta(x) \tag{2.19}$$

Normalizing Flow based generative models like Glow and Inverse autoregressive Flow are easier to parallelize for both training and synthesis. Normalizing Flow allows various applications such as interpolations between data points and meaningful modifications of existing data points. In this section, we will provide details of two normalizing Flow models, namely IAF and Glow, which are used as one of the deep learning components in this thesis work.

Inverse Autoregressive Flows

The inverse autoregressive Flow was introduced in 2017 [Kingma et al., 2016], as a way to scale well to high-dimensional latent spaces as well as allow faster inference. This family of Flow has a series of cascaded inverse autoregressive transformations. The architecture for the IAF model has three components, namely encoder, IAF Flow, and decoder. For given input x, encoder network generates an hidden output h. Then, hidden output h is given to the feedforward neural network to obtain the μ_0 and σ_0 as estimates of mean and standard deviation. The initial latent variable, z_0 is estimated by drawing random sample $\varepsilon \sim \mathcal{N}(0, I)$ for using the reparameterization as shown in Equation 2.20. Afterward, z_0

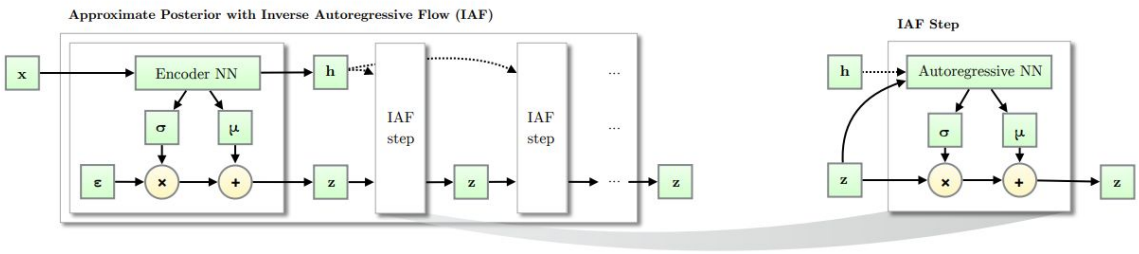

Figure 2.2 – Inverse autoregressive transformation cell from z_k to z_{k-1}, as a chain of nonlinear invertible transformations [Kingma et al., 2016]

along with hidden output h is provided to k steps of inverse autoregressive transformation to obtain flexible posterior probability distribution with latent variable z_k, refer Equation 2.21.

$$z_0 = \mu_0 + \sigma_0 \odot \varepsilon \tag{2.20}$$

$$z_k = \mu_k + \sigma_k \odot z_{k-1} \tag{2.21}$$

For each step of Flow transformation, the neural network model is designed to predict μ_k, and σ_k as shown in Figure 2.2. In Flow transformations, latent variable from the previous Flow step $k-1$ and hidden output from the previous Flow step are provided as an input. And hidden output h as input to autoregressive networks of Flow transformations. These autoregressive transformations are invertible if $\sigma_i > 0$ condition is satisfied for i^{th} value of D dimension [den Berg et al., 2018]. The autoregressive structure of Flow allows simple computation of the Jacobian determinant of each transformation as a change in global posterior probability density of encoder network denoted as $logQ(z_K|x)$, where z_K is the output of the last Flow step. Equation 2.22 provides a tractable change in probability density for which detailed derivation is provided in [Kingma et al., 2016]. In this way, the flexible, tractable posterior distribution is created to perform the variational inference with the inverse autoregressive Flow. Thus, the ability of flexible distribution to fit closely to the true posterior improves the performance of the autoregressive model and the depth of the chain.

$$logQ(z_K|x) = -\sum_{i=1}^{D}\left(\frac{1}{2}\varepsilon_i^2 + \frac{1}{2}log(2\pi) + \sum_{k=0}^{K} log(\sigma_{k,i})\right) \tag{2.22}$$

Inverse autoregressive Flow (IAF) models have been used previously in the context of speech processing applications. For fast and high-fidelity Wavenet based speech synthesis, the authors in [van den Oord et al., 2018] proposed probability density distillation to fill in the bridge between trained Wavenet as a teacher model and IAF as a student model. In [Esling et al., 2019], the authors proposed a universal audio synthesizer built using normalizing Flows to learn the latent space representation for semantic control of a synthesizer by interpolation of latent variables.

Generative Flow (Glow)

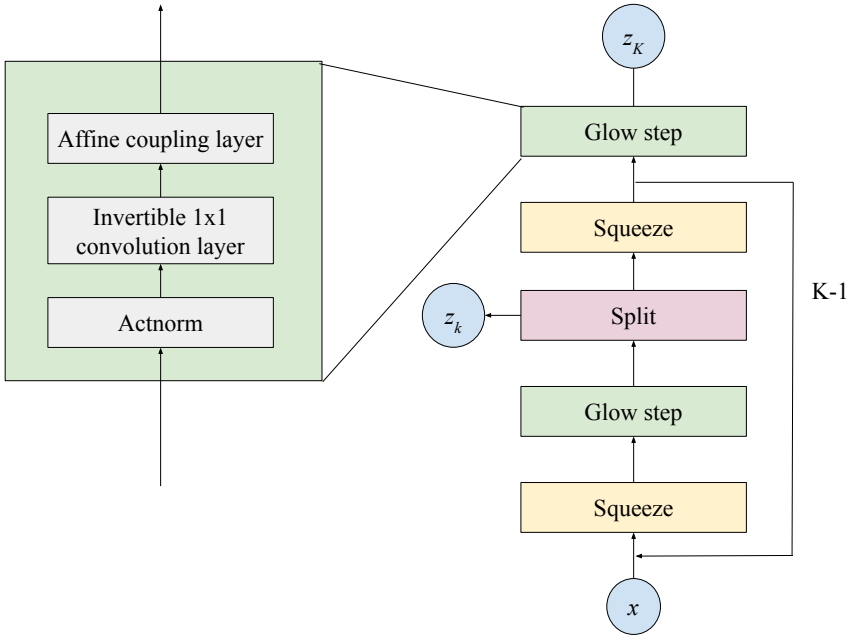

Figure 2.3 – Generative Flow where each transformation consists of Glow step (left), composed of an actnorm, followed by an invertible 1x1 convolution, and an affine transformation. Each Glow step is combined using multi-scale architecture (right).

Generative Flow (Glow) as generative model was proposed in [Kingma and Dhariwal, 2018] as an extension to non-linear independent components estimation [Dinh et al., 2015] and Real Non Volumetric Preservation (NVP) Flow [Dinh et al., 2017] using an invertible 1x1 convolution. Glow is a multi-scale architecture shown in Figure with Flow step cascaded transformation. Each Flow step is composed of an actnorm (activation normalization), followed by an invertible 1x1 convolution, then an affine coupling layer, refer to Figure 2.3. Like batch normalization, the actnorm (activation normalization) layer conducts affine transformation of activations using scale and bias parameters so that post-actnorm activations have zero mean and unit variance given an initial minibatch data. The actnorm layer provides the functionality of data-dependent initialization explained in [Kingma and Dhariwal, 2018], where scale and bias are trainable parameters. The usage of actnorm serves to reduce the internal covariant shift and provides stability during the training phase by avoiding vanishing or exploding gradients. The output of actnorm is given to a 1x1 invertible convolution layer designed to optimize the computing cost of the Jacobian determinant as a triangular matrix. Hence Equation 2.18 is reduced to Equation 2.23 as only diagonal components are considered for computing the determinant of a triangular matrix.

$$logP_\theta(x) = logP_\theta(z) + \sum_{k=1}^{K} log|diag\left(\frac{dz_k}{dz_{k-1}}\right)| \tag{2.23}$$

The concept of affine coupling layer was first introduced in [Dinh et al., 2017], where scale and translate vectors are computed using split and concatenation operations. The affine coupling layer of the Glow step utilizes any non-linear function without affecting the computation of log determinants of Jacobian, defined as diagonal components of scale. The design choice of additive coupling enables the usage of any arbitrary neural network to be used in computing scale and translate, explained in detail in [Kingma and Dhariwal, 2018]. The Glow model is optimized under the log-likelihood objective stated in Equation 2.19.

Glow has recently been used as a non-autoregressive model to synthesize speech waveform conditioned on acoustic features [Prenger et al., 2019, Kim et al., 2019]. The Glow architecture has shown state-of-art results in a neural vocoding task providing Mel spectrogram as additional input to affine coupling layer. After that, [Kim et al., 2020, Miao et al., 2020] presented as non-autoregressive TTS systems, where Glow architecture was used as a decoder network to generate Mel spectrogram from simple distribution.

2.3.3 Diffusion Probablistic Models

In 2015 diffusion probabilistic models introduced to model complex data distributions using stochastic calculus [Sohl-Dickstein et al., 2015, Popov et al., 2021]. The diffusion probabilistic models are inspired by non-equilibrium thermodynamics for systematically and slowly deconstructing the structure in a data distribution through an iterative forward diffusion process. After that, the reverse diffusion process learns to reconstruct the structure in data, creating a flexible and tractable generative model with an exact sampling scheme. In diffusion probabilistic models, the probability distribution is implicitly represented by the sampling procedure [Nichol and Dhariwal, 2021]. The denoising diffusion models have shown their ability to synthesize high-quality image samples better than previously published results for various generative models [Ho et al., 2020]. The denoising diffusion probabilistic model is developed by training the Markov chain for forward and reverse diffusions, thus leading to slower inference speed.

The score based generative model was proposed in [Song et al., 2021,Song and Ermon, 2020] to estimate the gradient of the log probability density function, called score function [Liu et al., 2016]. The score function is learned to match the probability density function through a neural network model. In [Ho et al., 2020] it is suggested that Markov chains are approximated trajectories of stochastic processes satisfying certain Stochastic Differential Equations (SDE). Equation 2.24 refers to SDE for diffusion probabilistic model, where the coefficient of diffusion, a and drift b is defined over W_t as a standard Brownian motion, $t \in [0, T]$, where T is some finite terminal time horizon. The stochastic calculus provides easy to find a stochastic process such that terminal distribution converges to standard normal distribution $\mathcal{N}(0, I)$. Therefore, for a given forward diffusion process, DPM models are optimized to reverse in time trajectories of the forward diffusion process with reverse diffusions.

$$dX_t = b(X_t, t)dt + a(X_t, t)dW_t \tag{2.24}$$

In such a scenario, the generation process boils down to sampling random noise from

$\mathcal{N}(0, I)$. Then, simple first-order Euler-Maruyama scheme [Kloeden and Platen, 1977] is used to solve SDE for describing dynamics of the reverse diffusions. If forward and reverse diffusion processes have close trajectories, then the distribution of resulting samples will be very close to that of the data. Recently proposed scoring function-based DPM model uses stochastic differential equations instead of Markov chains. Scoring function based DPMs allows forward diffusions to transform to any data distribution of $\mathcal{N}(\mu, \sum)$, and not necessarily $\mathcal{N}(0, I)$, where μ is mean and $\sum$ is diagonal covariance matrix.

$$dX_t = \frac{1}{2}\sum^{-1}(\mu - X_t)\beta_t d_t + \sqrt{\beta_t}dW_t \tag{2.25}$$

$$dX_t = \left(\frac{1}{2}\sum^{-1}(\mu - X_t) - \nabla logp_t(X_t)\right)\beta_t dt + \sqrt{\beta_t}d\tilde{W}_t \tag{2.26}$$

For n-dimensional stochastic process X_t, SDE can be defined as Equation 2.25 for transforming any data to Gaussian noise over finite time horizon T and noise schedule β_t as forward diffusion process. Thereafter, SDE for reverse diffusion can be defined as Equation 2.26 for reverse time dynamics of stochastic process of diffusion's obtained from [Anderson, 1982]. In Equation 2.26 $d\tilde{W}_t$ is reverse-time Brownian motion, and p_t is the probability density function of random variable X_t, where $t \in [0, T]$ to solve SDE backward in time. In score based model [Song et al., 2021] SDE Equation 2.26 can be assumed to be ordinary differential Equation, and thus Equation 2.26 can be rewritten as stated in Equation 2.27, where $\nabla logp_t(X_t)$ represent an estimate of gradient of the log density of noisy data as score function of distribution $P(X)$, where $P(X) = P(X_t|X_0)$, as probability distribution conditioned on initial distribution of $P(X_0)$.

$$dX_t = \left(\frac{1}{2}\sum^{-1}(\mu - X_t) - \nabla logp_t(X_t)\right)\beta_t dt \tag{2.27}$$

$$dX_t = \left(\frac{1}{2}\sum^{-1}(\mu - X_t) - s_\theta(X_t, \mu, t)\right)\beta_t dt \tag{2.28}$$

$$\mathcal{L}_{Fisher-divergence} = \mathbb{E}_{p(x)}[\|\nabla logp_t(X_t) - s_\theta(X_t, \mu, t)\|_2^2] \tag{2.29}$$

According to forward Kolmogorov equations, Equation 2.25 and Equation 2.27 presents the evolution of probability density functions of an identical stochastic process. Therefore, score based DPM model learn to approximate neural network $s_\theta(X_t, \mu, t)$ such that s_θ is approximate estimate of $\nabla logp_t(X_t)$, which reduces Equation 2.27 to Equation 2.28. During the training of the DPM model, Fisher divergence is used to minimize loss criterion between model and noisy data distribution defined as in Equation 2.29. A more detailed description of the derivation of the DPM model used in the scope of thesis work can be found in [Sohl-Dickstein et al., 2015, Ho et al., 2020, Song et al., 2021].

Scoring-based DPM architectures have been proposed for speech synthesis, such as neural vocoders [Chen et al., 2021a,Chen et al., 2021b,Kong et al., 2021] with performance matching to state-of-art deep generative models. Grad-TTS [Popov et al., 2021], and Diff-TTS [Jeong et al., 2021] were recently published for the TTS task where decoders are designed using diffusion probabilistic models. These proposed TTS architectures provided

a flexible inference scheme with the trade-off in terms of the quality of synthesized speech and inference speech.

2.4 Deep Metric Learning

Metric learning has caught the interest of researchers looking for ways to extract discriminative features in transformed data for statistical models as well as deep neural network models. The metric learning approach takes into account the similarity or dissimilarity of data points. The goal of metric learning is to minimize the distance between objects of the same class and widen the gap between items of different classes [Kaya and Bilge, 2019]. Deep learning architectures enable the learning of global features from input data to reduce the feature dimensionality for extracting discriminative features. The reduced representation of input data allows understanding latent semantic information as vital to discriminative tasks. Therefore, the application of metric learning in deep learning architectures provides an enhanced representation of discriminative features. The classical usage of metric learning is in face verification and recognition [Liu et al., 2015, Hu et al., 2014, Schroff et al., 2015]. Additionally, deep metric learning has been widely applied for video understanding [Huang et al., 2018], speaker verification [Chen et al., 2019], speaker diarization [Narayanaswamy et al., 2019], medical problems [Litjens et al., 2017], signature verification [Bromley et al., 1993], music similarity [Lu et al., 2017] etc.

Deep metric learning consists of three main parts, which are input data, the structure of the deep neural network, and a metric loss function. After providing input data to a deep neural network, an intermediate latent embedding is extracted. Furthermore, latent embedding and reference latent embeddings are given to the metric loss function, thus measuring the similarity distance focused on desired semantic information.

The contrastive loss has been used to improve the performance of the Siamese network, where shared weights of the neural network model are updated according to the loss criterion [Hadsell et al., 2006, Jeong et al., 2018]. The purpose of the loss criterion is to minimize the distance between latent embeddings of data points of the same class and increase the distance between latent embeddings of data points of different classes. The contrastive loss used in Siamese neural network is described by Equation 2.30, where m is margin value, and $\mathcal{Y}$ is a binary class value equal to 0 if pair of input z_1, z_2 are from the same class or 1 otherwise.

$$\mathcal{L}_{contrastive} = (1 - \mathcal{Y})\frac{1}{2}\|z_1 - z_2\|_2 + (\mathcal{Y})\{max(0, m - \|z_1 - z_2\|_2)\} \tag{2.30}$$

A neural network with triplet loss inspired by the Siamese network contains three latent embeddings, which are positive, negative, and anchor [Hoffer and Ailon, 2015]. Triplet loss was proposed to minimize L_2 distance between anchor, z_{anchor} and positive, $z_{positive}$ latent embeddings while maximizing L_2 distance between latent embeddings of anchor, z_{anchor} and negative, $z_{negative}$ classes as stated in Equation 2.31. The triplet loss only considers one negative and one positive latent embedding to be compared with anchor training samples in a single backpropagation pass using shared weights of the neural network. In [Wang et al., 2017a], a novel angular loss was proposed to consider angular information

for comparison purposes. Unlike Siamese network and triplet loss, angular loss focuses on angular constrain at the negative latent embedding of triplet triangles, as stated in Equation 2.32, where α is the marginal angular constraint. The angular loss pushes the negative representations away from the center of the positive cluster. Additionally, the usage of angle as a similarity measure provides scale-invariant and rotation metrics.

$$\mathcal{L}_{triplet} = max(0, \|z_{anchor} - z_{positive}\|_2 - \|z_{anchor} - z_{negative}\|_2 + m) \tag{2.31}$$

$$\mathcal{L}_{angular} = max(0, \|z_{anchor} - z_{positive}\|_2 - 4tan^2\alpha\|z_{negative} - \frac{z_{anchor} + z_{positive}}{2}\|_2) \tag{2.32}$$

The contrastive loss and triplet loss often suffer from slow convergence. In the training phase, both losses utilize only a single negative latent embedding without interaction with other negative classes. Multi-class N-pair loss was proposed to address this problem by considering latent embeddings from all negative classes [Sohn, 2016]. Therefore, multi-class N-pair loss increases the intercluster distance from $N-1$ negative latent embeddings out of N total classes. And decreases the intracluster distance between positive latent embeddings, as described by Equation 2.33. The usage of multi-class N-pair loss has shown improved results in various computer vision applications such as visual recognition, and verification, object clustering.

$$\mathcal{L}_{N-pair} = log(1 + \sum_{i=1}^{N-1} e^{z_{anchor} \cdot z_{negative,i} - z_{anchor} \cdot z_{positive}}) \tag{2.33}$$

2.5 Speech

A speech waveform is a time-varying representation of intensity characterized by amplitude. Speech waveforms are typically handled in digital form, and speech processing can be considered as the junction of digital signal processing and natural language processing [Hassanien et al., 2008, Deller and Hansen, 2005]. Before applying any computation algorithm to the speech signal, the analog representation of the speech waveform should be transformed into a representation comprehensible by digital processing devices. As a result, an analog-to-digital converter is used to sample the analog speech signal recorded through the microphone to a discrete signal. Speech processing applications have traditionally used features derived from digitalized speech waveforms.

The time-domain representation of speech focuses solely on intensity and ignores frequency domain information. The speech waveform is converted into a frequency-domain representation to better comprehend various frequency components. The short-time Fourier-transform (STFT) is used to convert it into a frequency domain representation with regular intervals of time windows. The use of short-time windows enables the analysis of a signal's attributes at a specific time frame. Equation 2.34 defines the STFT of a speech signal x_n and a window w_n over N samples, as $X_{t,f}$. Since the STFT transformation provides a complex-valued feature, $20log_{10}(|X_{t,f}|)$ is calculated for visual representation,

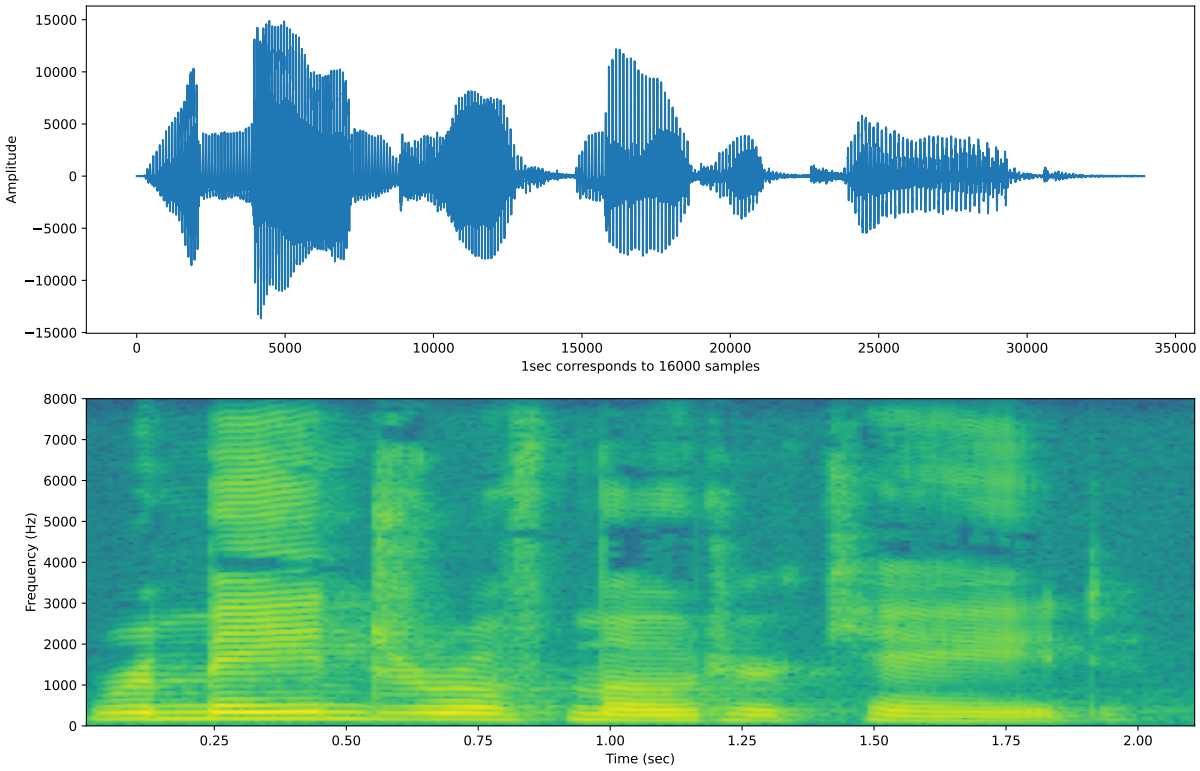

Figure 2.4 – Speech waveform representation (top) in time domain with sampling rate of 16000; Short-time-fourier-transform (STFT) (bottom) for representing speech waveform in time-frequency domain.

commonly known as a spectrogram. The time-frequency domain representation of the speech waveform is shown in Figure 2.4.

$$X_{t,f} = \sum_{n=0}^{N-1} x_{n+t} w_n e^{-i2\pi \frac{kn}{N}} \tag{2.34}$$

Stevens, Volkmann, and Newmann proposed the Mel scale in 1937 to make the pitch of speech waveform sound equally perceptual to the listener [Stevens et al., 1937]. The Mel scale is a perceptual scale of pitch that effectively reduces frequency resolution as the frequency increases. The Mel spectrogram is created by transforming the spectrogram's frequencies to the Mel scale. Equation 2.35 defines the mathematical function for converting frequencies, f, to Mel frequencies, m. Figure 2.5 [1] depicts the Mel scale curve as a frequency variation. Using such an auditory frequency scale has the effect of emphasizing details in lower frequencies, which are critical to speech intelligibility [Shen et al., 2018]. Mel spectrogram is widely used as acoustic features in various speech applications such as speech recognition, speaker verification, speech synthesis, etc.

1. Image Source:https://www.researchgate.net/figure/pitch-scale-vs-mel-scale_fig4_350128340

$$m = 2595 \cdot log\left(1 + \frac{f}{700}\right) \tag{2.35}$$

Figure 2.5 – Mel scale plot stating equally spaced intervals of frequencies perceived by human ear, where horizontal axis are linear Hz frequencies, and vertical axis are Mel frequencies.

2.6 TTS: State of the art

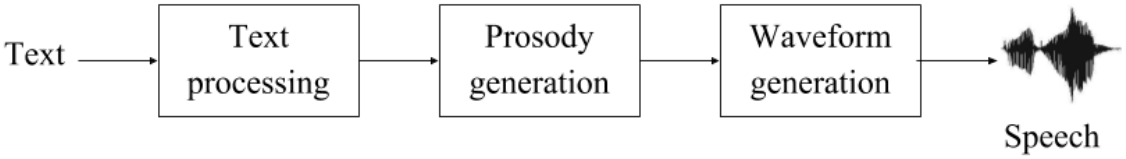

Figure 2.6 – Framework for speech synthesis system

The general framework for speech synthesis systems includes text analysis, prosody generation, and waveform generation, as shown in Figure 2.6. The text analysis is a crucial step in speech synthesis, which extracts linguistic, phonetic, and prosodic information from the input text and creates a representation of text to predict prosody for the generation of speech. This linguistic representation drives the synthesis routines to generate the speech waveform of the input text [Klatt, 1980]. In the text analysis phase, grapheme-to-phoneme conversion is included and is still used in modern end-to-end TTS systems. After that, output from text analysis is processed through prosody generation to predict the acoustic features relevant for speech waveform generation. In this Section 2.6, we will overview various techniques used to develop TTS systems. Prosody parameters refer to aspects of the speech signal at a level above the individual phoneme, such as intonation, rhythm, and intensity.

2.6.1 Articulatory Synthesis

An articulatory synthesis provides a better understanding of the acoustic influence of speech articulator motions by connecting the acoustic and articulatory domains of speech. Therefore, the speech production approach is vital in apprehending the principles behind articulatory synthesis [Elie and Laprie, 2016]. Articulatory speech synthesis models incorporate rules derived from human articulators and vocal cords to generate speech waveform. Articulatory models are complex systems to implement with a considerable amount of computational complexity [Liberman et al., 1959,Rahim et al., 1993]. In 1987, Klatt proposed the first articulatory model based on vocal tract area as a function of the larynx to lips for each phonetic segment [Klatt, 1987].

The rule-based articulatory synthesis systems have been proposed to control the parameters conditioned on phonemes such as glottal aperture, cord tension, and lung pressure [Liberman et al., 1959]. During speech production, vocal tract muscular movements result in movement and change of the volume in articulators, resulting in various speech sounds. The articulatory models are designed by analyzing X-ray or MRI data of speaker recording. Articulatory models face several obstacles in development due to insufficient availability of articulatory data as well as difficulties in modeling 2D vocal tract data to 3D volumetric data [Douros, 2020].

2.6.2 Formant Synthesis

The term formant is attributed to distinctive (high energy) frequency components in the speech that correspond to the acoustic resonance of the human vocal tract. The formant-based speech synthesis method is constructed with the help of the physical and spectral models as a source-filter model. In this approach, the source models the glottal pulse, which is then passed through filters to imitate the human vocal tract [Schröder, 2009]. The synthetic speech is produced with rules to define prosody parameters such as fundamental frequency, intensity, rhythm over time. Due to limitations of the rule-based system, formant synthesis is not able to generate a natural voice.

Various formant speech synthesis systems have been proposed depending on the structure of formant resonators. A cascade formant synthesis is mainly composed of band-pass resonators connected in series. The output of each formant resonator is passed through a cascaded structure, which requires the formant frequencies as additional input. One advantage of such a cascaded structure is that no explicit adjustment is required for tuning the formant amplitudes for vowels. [Allen et al., 1987]. Contrary to cascaded structure, parallel formant synthesizer organizes formant resonators in a parallel fashion. In addition to parallel resonators, nasal resonators are also incorporated in parallel formant synthesizers. For a given excitation signal to parallel resonators, outputs of each resonator are summed as an output speech. According to Saughnessy, [O'Shaughnessy, 1987], adjacent outputs of formant resonators must be summed in opposite phases to avoid undesirable zeros or anti-resonances. The parallel structure of the formant synthesizer eases the explicit control over bandwidth and gains for each formant resonator.

Dennis Klatt proposed a more complex formant synthesizer in 1980, which included extra resonances and anti-resonance for nasalized sounds, the sixth formant for high-

frequency noise, a bypass circuit to create a flat transfer function, and radiation characteristics. The system used a complicated excitation model with 39 parameters that were updated every five milliseconds [Klatt, 1980]. The formant synthesis does not rely on a vast database of speech data and does not require pre-recorded speech for synthesis. As a result, formant synthesis can be deployed in embedded systems where memory is a critical consideration.

2.6.3 Unit Selection

The first data-driven approach relied on a concatenation of diphones to construct a speech waveform for a specific language at the beginning of the 1990s [Schröder, 2009]. This method was further improved by the use of diverse diphone unit samples recorded in natural speech. The diphone unit is a speech segment from the middle of phone to the middle of the next one. Unit selection method generates synthesized speech using selection and concatenation of units (diphones, triphones) from speech database recorded with variation in units for prosody features and duration.

In unit selection synthesis, diphone units are selected based on optimal path algorithms such as the Viterbi algorithm for a given text. In this method, an appropriate (optimal) list of units is selected using both concatenation scores (how well two adjacent units concatenate) and target scores (how well units match target criteria); therefore, this method is called the unit selection. As the unit selection method primarily depends on recorded speech, for expressive speech synthesis, diphone units must be recorded for each emotion [Fernandez and Ramabhadran, 2007]. A given emotion is then generated by selecting units only from the corresponding subset of the recordings. The large databases of sound units with variations in prosody and spectral features enable the development of more natural-sounding speech than the usage of a small amount of sounds units [Campbell and Black, 1997].

2.6.4 Statistical Parametric Speech Synthesis

The application of Hidden Markov Model (HMM) is not new in speech processing, as it was widely used not only in speech recognition systems. But also to build speech synthesis systems, which made HMM the most suitable model in the speech processing domain [Rabiner, 1990]. The Hidden Markov Model (HMM) technique was used for the speech synthesis system, predicting acoustic and duration parameters from statistical models extracted from recorded speech.

For speech recognition, the context is mainly restricted to previous and following phones; the context includes much more linguistic, phonetic, and prosodic information in speech synthesis. In this approach, the densities of context-dependent models are organized in a decision tree; At run-time, for a given "target" context to be realized, the tree yields the appropriate sequence of probability densities. These corresponds to context, describing the mean and standard deviation of the acoustic features [King, 2010, Zen et al., 2009]. Each HMM has state duration densities to model the temporal structure of speech. This approach is called statistical as it represents the parameters using means

and variances of probability density function estimated from training data. The approach is also called parametric as a set of parameters are used to describe the speech signal.

A synthesized utterance is converted to a context-dependent label sequence for a given text. Then the utterance HMM is constructed by concatenating the context-dependent HMMs according to the label sequence. Secondly, state durations of the HMM are determined based on the state duration probability density functions [Tao et al., 2014]. Like unit selection based on expressive speech synthesis, the simple solution is to train the HMM model on a speech dataset with various emotions and speaking styles. The further strength of HMM-based synthesis is that speaker-specific or style-specific voices can also be created by adaptation rather than training [Schröder, 2009].

2.6.5 Neural Speech Synthesis

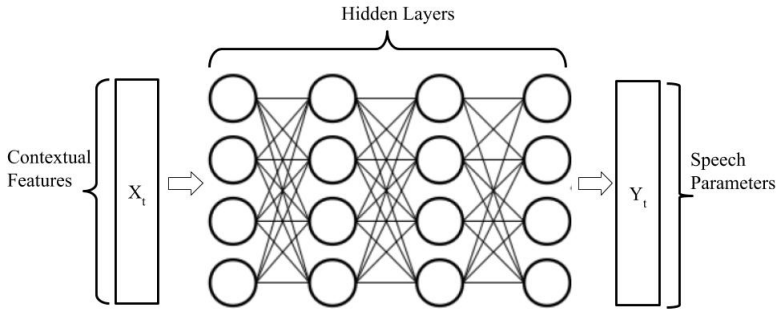

Figure 2.7 – Deep neural network architecture for speech synthesis.

Due to a lack of computational resources, a small amount of training data, and efficient training methods in the early 1990s, neural network approaches were unable to gain attraction [Cawley and Noakes, 1993, Tuerk and Robinson, 1993]. In neural speech synthesis, HMM and their associated decision trees are replaced with a feedforward neural network, which predicts the speech parameters directly from the set of contextual labels. The speech waveform is constructed from these predicted acoustic features [Zen et al., 2013], [Wu et al., 2016].

The neural network approach enhances the ability of the model to map the complex non-linear relationship between context-dependent decision trees to vocoder parameters as a multi-dimensional non-linear regression problem. A neural network parametric TTS system is usually composed of a duration model and an acoustic model constructed with neural network layers. Many approaches have been proposed to use DNN to develop parametric TTS systems [Zen et al., 2013, Lu et al., 2013, Qian et al., 2014, Wu et al., 2015, Hashimoto et al., 2015] and expressive parametric TTS systems [Yamagishi et al., 2004, Xue et al., 2018, Li et al., 2018].

Before estimating acoustic features using neural network models, the text is first converted into phonemes. Subsequently, these phonemes are mapped to input contextual features (conveying phonetic, linguistic, and prosodic information) denoted as X_t. The input contextual features are transformed into answers about linguistic information such

as whether the current phoneme is a vowel or not, position information about the current phoneme in a sentence, etc. The neural network predicts the speech parameters, Y_t for a given input contextual features for each frame as shown in Figure 2.7. Finally, the speech waveform is synthesized using a vocoder with the predicted acoustic features.

Few strategies for transferring expressivity to a new speaker's voice in a parametric TTS framework have been published [Parker et al., 2018, Chen and Braunschweiler, 2013, Chen et al., 2015, Ohtani et al., 2015]. Expressivity transfer is accomplished in [Ohtani et al., 2015] by creating an Eigen voice space of neutral speech as well as the contrasts between neutral and expressive speech. Two separate clusters adaptive training (CAT) subspaces are produced in [Chen and Braunschweiler, 2013, Chen et al., 2015], one for speakers and the other for expressiveness. The expressive utterance is projected into the expression subspace to gain the appropriate expressive CAT weights. The neutral utterance is subsequently projected into the speaker subspace to gain the necessary speaker CAT weights.

2.7 Vocoder

Vocoders play a vital role in developing text-to-speech synthesis for producing naturally sounding speech from predicted acoustic features. In the context of speech synthesis, vocoders can be classified into two categories: digital signal processing and deep neural networks. Most prominently used vocoders in parametric TTS systems are STRAIGHT [Kawahara et al., 2001] and WORLD [Morise et al., 2016]. Vocoders provide two functionalities, speech analysis and speech synthesis. The speech analysis step is used for extracting acoustic features such as Mel cepstral coefficients, band aperiodicity, fundamental frequency, and voiced-unvoiced information. Conversely, the speech synthesis step generates speech waveform from features extracted from the speech analysis.

In 2016, Aaron and his colleagues introduced a neural vocoder called Wavenet, which has shown outstanding performance in synthesizing natural-sounding speech compared to traditional vocoders [Oord et al., 2016]. Wavenet is an autoregressive deep generative model that combines causal filters with dilated convolutions. The usage of dilated convolutions leads to an increase in the receptive fields exponentially with the depth of the neural network model. As a result, the Wavenet model was able to learn long-range temporal dependencies. The originally proposed Wavenet architecture suffers from slow inference speed, which was improved in architecture proposed by the same authors in Parallel Wavenet [van den Oord et al., 2018]. Probability density distillation is used for reducing the divergence between output distribution of trained Wavenet as a teacher model and inverse autoregressive Flow as a student model for fast and high-fidelity Wavenet-based speech synthesis. The IAF transforms the sequence of noise into a speech waveform without an autoregressive generation. Many variants of Wavenet have been proposed over the years such as WaveRNN [Kalchbrenner et al., 2018], WaveGAN [Donahue et al., 2019], WaveGlow [Prenger et al., 2019], Nv-WaveNet [Davis et al., 2020], FFTNet [Jin et al., 2018], LPCNet [Valin and Skoglund, 2019].

2.7.1 WORLD Vocoder

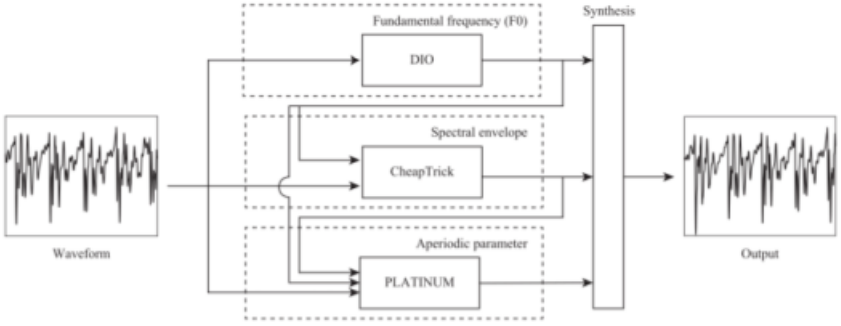

Figure 2.8 – WORLD consists of three analysis algorithms for determining the F0, spectral envelope, and aperiodic parameters and a synthesis algorithm [Morise et al., 2016].

In 2009, Masanori Morise developed the WORLD vocoder, which is open source and freely available [Morise et al., 2016]. It consists of three speech analysis modules to compute spectral envelope, band aperiodicity, and fundamental frequency (F0), as depicted in Figure 2.8. The WORLD vocoder estimates the F0 contour with DIO F0 estimation algorithm [Morise et al., 2009]. The spectral envelope is estimated using the CheapTrick algorithm, which uses speech waveform and the F0 information [Morise, 2015]. Afterward, aperiodicity in the waveform is estimated with the PLATINUM method and aperiodic parameter extraction algorithm [Morise, 2012]. The performance of the WORLD vocoder illustrates the ability to synthesize high-quality speech in real-time processing.

2.7.2 WaveGlow

WaveGlow is a non-autoregressive model which combines Glow and Wavenet as a neural vocoder to convert Mel spectrogram into raw speech waveforms [Prenger et al., 2019]. WaveGlow is a deep generative model that generates speech waveform by sampling latent variables from a multivariate Gaussian distribution with zero mean and unit variance. The WaveGlow model is trained with a single loss criterion for minimizing the log-likelihood of the training data distribution, which is also explained in Section 2.3.2.

The bijective design of each Glow transformation ensures invertibility and tractability in the Glow-based generative model. The Glow transformation blocks of WaveGlow use a 1x1 invertible convolution layer, an affine coupling layer, and a squeeze operation to construct a series of invertible transformations. During the training phase, the Mel spectrogram is firstly upsampled to match the sampling rate of speech and passed through non-linear transformation by the Wavenet network. The Wavenet architecture is similar to the one used in [Oord et al., 2016] with modification of a non-causal dilated convolutional layer. As the affine coupling layer maintains the invertibility of the network, and only scale term is incorporated in computing change in volume in Glow transformation, this allows the usage of Wavenet as any arbitrary function [Dinh et al., 2017, Kingma and

Dhariwal, 2018]. Speech waveforms are directly synthesized by conditioning Glow transformation on the Mel spectrogram. WaveGlow provides a fast, efficient, and high-quality speech synthesis model, with mean opinion scores similar to Wavenet vocoder.

2.7.3 Hi-Fi GAN

The High Fidelity (Hi-Fi) GAN was introduced as a neural vocoder with matching performance to counterparts based on Autoregressive models [Oord et al., 2016], and Normalizing Flow-based models [Prenger et al., 2019]. One of the salient features of Hi-Fi GAN is the usage of sinusoidal signals with various frequencies and modeling the periodic patterns to generate realistically, high-quality speech waveform [Kong et al., 2020]. Furthermore, Hi-Fi GAN utilizes multiple residual blocks to extract the patterns of variable lengths of sinusoidal signals and gives them to the generator module of the GAN network. Hi-FI GAN has shown better performance than the publicly available WaveGlow and Wavenet models in MOS scores.

HiFi-GAN is composed of the generator and two discriminators, namely multi-scale discriminators and multi-period discriminators. The generator is a fully transposed convolutional neural network. It uses a Mel spectrogram as input to upsample using transposed convolutions until the length of the output sequence matches the temporal resolution of speech waveforms. Every transposed convolution is followed by a multi-receptive field fusion module, which observes patterns of various lengths in parallel. The multi-receptive field provides output as the sum of outputs from multiple residual blocks.

The multi-period discriminator is composed of sub-discriminators stacked together. Each sub-discriminator focuses on the different parts of the input speech waveform. The sub-discriminators are implemented using a strided convolutional layer with a leaky ReLU activation function proceeded by weight normalization.

To ensure the stable training of Hi-Fi GAN, generators and discriminators are trained with two additional losses, namely, Mel spectrogram loss and feature matching loss. The Mel spectrogram loss is the L_1 distance computed between the Mel spectrogram of speech synthesized by the generator and the Mel spectrogram of the original speech waveform. The second loss criterion is feature matching loss, which measures the similarity by estimating the difference in embeddings of the discriminator between synthesized speech and original speech [Larsen et al., 2016].

2.8 End-to-end TTS

The progress in text-to-speech is no exception to deep learning, thus leading to deep neural network architecture trained as a single TTS system called end-to-end TTS. Since the introduction of SampleRNN [Klejsa et al., 2019], and Wavenet [Oord et al., 2016], a tremendous amount of research work has been published focusing on end-to-end TTS systems. Even though SampleRNN is an unconditional audio generation model, the addition of attention-based duration modeling and F0 prediction neural model leads to Char2Wav [Sotelo et al., 2017] as a raw speech waveform synthesis TTS system. Char2Wav depends on WORLD vocoder features for generating pretrained alignment for

auxiliary modules such as F0, spectral envelope, and aperiodicity features. The Wavenet model presented in [Oord et al., 2016] Wavenet model is locally conditioned on linguistic features derived from input text for the text-to-speech synthesis task.

The proposed end-to-end TTS systems in [Shen et al., 2018, Wang et al., 2017b, Arik et al., 2017, Ren et al., 2019, Sotelo et al., 2017] have done pioneering work in training complete TTS system, which is then preceded by improving existing architectures of end-to-end TTS systems [Gibiansky et al., 2017, Ping et al., 2017, Ren et al., 2020a, Elias et al., 2021a, Elias et al., 2021b]. However, these systems still has a bottleneck step of vocoder, as input-output of end-to-end TTS is <text, Mel spectogram>. Few approaches have been proposed to resolve bottleneck neural vocoder step jointly training neural vocoder alongside the Mel spectrogram predicting module [Oord et al., 2016, Sotelo et al., 2017, Weiss et al., 2021], which is still limited to single-speaker neutral TTS framework.

In 2017, Yuxuan Wang and his colleague presented Tacotron architecture based on sequence to sequence learning with attention mechanism [Wang et al., 2017b]. The Tacotron speech synthesis system adopts a multi-stage encoder-decoder architecture with multiple RNNs and blocks called CBHG (convolutional block highway gated recurrent units). Each CBHG contains multiple convolutional layers, a highway network, and a bidirectional GRU. The Griffin-Lim method generates the speech waveform from output synthesized spectrogram, where acoustic features are directly conditioned on input text instead of linguistic features, making Tacotron a complete end-to-end system without explicit knowledge of a language.

Over the years various approaches have been proposed as an extension to Tacotron architecture such as Tacotron 2, parallel Tacotron, parallel Tacotron 2 [Shen et al., 2018], Wave Tacotron [Weiss et al., 2021], non-attentive Tacotron [Shen et al., 2020], Flowtron [Valle et al., 2021], Mellotron [Valle et al., 2020]. The Tacotron architecture implicitly learns the duration information through location-sensitive attention. Non-autoregressive approaches presented in non-attentive Tacotron, parallel Tacotron, and parallel Tacotron 2 explicitly model the duration with the help of auxiliary loss computed from duration predictor. The Wave-Tacotron removes the bottleneck vocoder step by incorporating Generative Flow coupled with an autoregressive decoder to synthesize speech waveform directly. In addition to this, several other variants of autoregressive end-to-end TTS have been proposed such as deep voice [Arik et al., 2017], Clarinet [Ping et al., 2019], ParaNet [Peng et al., 2020], transformer TTS [Li et al., 2019], deep convolutional TTS DCTTS [Tachibana et al., 2018] constructed with convolutional neural networks.

Recently proposed Transformer TTS replaces RNN based encoder-decoder-attention model with multi-head attention in sequence to sequence learning framework. Transformer TTS proposed novel scaled positional embeddings in addition to phoneme embeddings, which enables the transformer encoder network to comprehend the order of phoneme sequence. Due to the removal of recurrent decoder architecture, Transformer TTS provides faster training and inference time than the Tacotron 2 [Li et al., 2019]. Fastspeech as a non-autoregressive TTS was proposed to provide an improvement over transformer TTS. Fastspeech utilizes a feed-forward neural network for transformer network to predict the Mel spectrogram [Ren et al., 2019]. Fastspeech TTS incorporates a teacher-student knowledge distillation approach for learning the phoneme duration and length-regulator from pretrained TTS models. Due to the prerequisite of knowledge distillation, it is a

time-consuming and complex process to train the Fastspeech TTS. After that, Fastspeech 2 presented two significant extensions to existing Fastspeech: removing knowledge distillation and introducing variance adapter to augment latent information of duration, pitch, and energy.

Non-autoregressive systems have shown speech synthesis performance similar to autoregressive counterpart, namely, Talknet [Beliaev et al., 2020], Talknet 2 [Beliaev and Ginsburg, 2021], Flow-TTS [Miao et al., 2020], Glow-TTS [Kim et al., 2020], Grad-TTS [Popov et al., 2021], Diff-TTS [Jeong et al., 2021], Align-TTS [Zeng et al., 2020]. In Talknet TTS architecture, alignment between phoneme sequence and Mel spectrogram is learned through pretrained connectionist temporal classification (CTC) trained on a speech recognition system. Talknet 2 TTS system has a pitch predictor module on top of the duration predictor and Mel generator network, thus providing more controllability over synthesized speech. Glow decoder has been proposed as a Mel spectrogram generator in Flow-TTS, Glow-TTS. The significant difference between Flow-TTS and Glow-TTS is the way phoneme duration and alignment are estimated. In the case of Flow-TTS, positional attention was employed to learn the alignment and phoneme duration. At the same time, Glow-TTS used explicit duration predictors and monotonic alignment similar to the Viterbi algorithm. Diffusion probabilistic models' recent success as a deep generative model paved the way for their application as Mel spectrogram generator networks. The decoder network of Grad-TTS and Diff-TTS is based on diffusion probabilistic models with duration predictors. In this thesis work, we proposed to extend the Tacotron 2, Glow-TTS, and Grad-TTS systems. Now, we will overview the detailed architectures of Tacotron 2, Glow-TTS, and Grad-TTS.

2.8.1 Tacotron 2

Tacotron 2 TTS system consists of two modules, first a recurrent sequence to sequence Mel spectrogram generator network with attention, where the sequence of Mel spectrograms is predicted from the input sequence of characters, as depicted in Figure 2.9. Afterward, Wavenet is used as a neural vocoder to generate speech waveform from predicted Mel spectrogram frames.

The Mel spectrogram generator network is comprised of an encoder, decoder, and attention. The encoder network first takes input as a sequence of characters mapped from one-hot representation to learnable character embedding of dimension 512. After that, character embeddings are passed through three convolutional layers of 512 filters and a kernel size of 5x1, followed by batch normalization and ReLU activation function. The kernel size of 5x1 aids the network in understanding the contextual information from a sequence of characters. Finally, the output of the convolutional layer is given to the Bidirectional LSTM layer with 512 hidden units to generate the encoder output representation.

Tacotron 2 utilizes location-sensitive attention [Chorowski et al., 2015] which is cumulative attention weights from previous decoder steps as an additional input to the attention network. The attention probabilities are estimated from the linear projection of encoder output and output of location-sensitive attention. Tacotron 2 decoder is implemented as an autoregressive recurrent neural network for predicting Mel spectrogram frame by

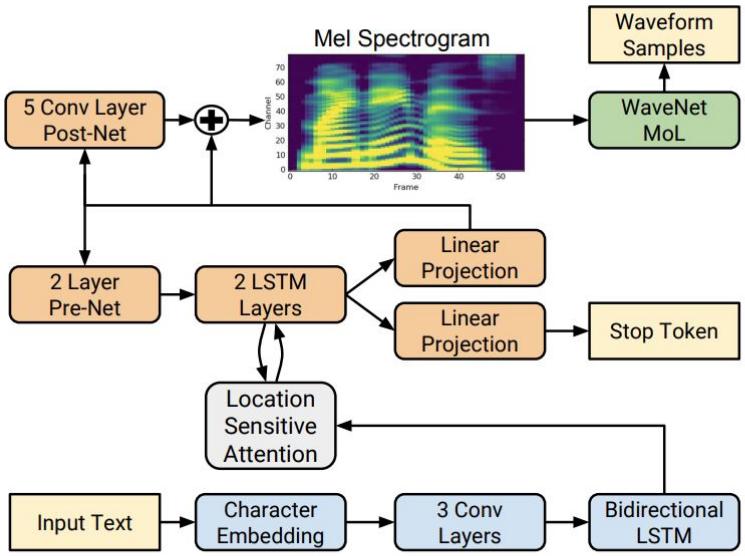

Figure 2.9 – Block diagram of the Tacotron 2 system architecture [Shen et al., 2018]

frame. The pre-net network is used for providing unsupervised guidance through previously predicted frames of the Mel spectrogram, thus ensuring the learning of attention as an alignment. The pre-net network is composed of 2 feedforward layers of 256 hidden units with a ReLU activation function. The output from the pre-net and context vector from the attention network is provided to two unidirectional LSTM layers of 1024 hidden units.

The output of LSTM layers is linearly projected and passed through the sigmoid activation function to estimate the probability of the output sequence as a stop token. Therefore, Mel spectrogram generation is completed when the output of the sigmoid function exceeds the 0.5 value. The output of linear projection is passed through the post-net network to estimate the residual to add to the predicted Mel spectrogram. Hence, post-net assists in improving the overall reconstruction performance of the Tacotron 2 system. The post-net network is composed of 5 convolutional layers with 512 hidden units and a kernel size of 5x1 with batch normalization and Tanh activation function except the last layer. The Tacotron 2 network is trained using mean square error loss criteria applied to the input of the post-net network and output of the post-net network. The Tacotron 2 TTS system is trained as an end-to-end TTS system without the explicit need for rule-based feature engineering and provides state-of-the-art quality of synthesized speech close to that of human speech.

2.8.2 Glow-TTS

Glow-TTS was presented in [Kim et al., 2020] as a non-autoregressive TTS based on the Generative Flow model. Glow-TTS learns the alignment between text and Mel spectrogram internally by explicitly modeling phoneme duration information. Glow-TTS

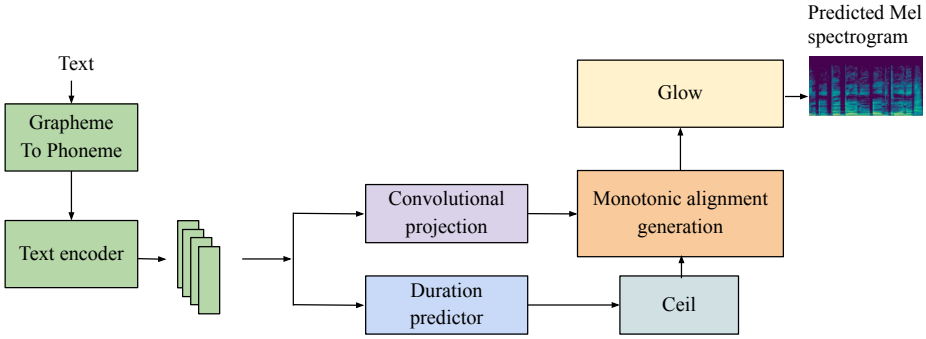

Figure 2.10 – Block diagram of the Glow-TTS system architecture

compounds the properties of Normalizing Flow and dynamic programming. Glow-TTS is composed of a text encoder, duration predictor, monotonic hard alignment, and decoder network, as illustrated in Figure 2.10. Monotonic hard alignment between text and the latent representation of speech is learned without an external pre-trained aligner. The text encoder of Glow-TTS utilizes the same model architecture as used in Transformer TTS [Li et al., 2019] with the modification of replacing positional encoding with relative position representation. Furthermore, residual connections are incorporated into the pre-net of the text encoder. The duration predictor is implemented using two convolutional layers with ReLU activation, layer normalization, and dropout layer, which is the same as used in FastSpeech TTS [Ren et al., 2019].

The core idea of Glow-TTS is to transform the conditional prior distribution $P_Z(z|c)$ to $P_X(x|c)$ using Glow as decoder, $f_{Glow} : z \rightarrow x$, where x is Mel spectrogram, z is latent variable and c is text. Therefore, the exact log-likelihood estimate can be stated using Equation 2.18 using the change of variable theorem. The conditional prior distribution is parameterized using alignment function A and network parameter θ. Therefore, the prior distribution P_Z is the isotropic Gaussian multivariate distribution, and the statistics of prior distribution are derived from a text encoder. The text condition $c_{1:T_{text}}$ of length T_{text} is mapped to $\mu_{1:T_{text}}$, and $\sigma_{1:T_{text}}$ with the help of text encoder. The alignment function A is modified Viterbi algorithm [Rabiner, 1989] implemented as monotonic search alignment; for more details, refer [Kim et al., 2020]. Alignment A is trained to maximize the log-likelihood of the most probable alignment between statistics of text encoder and sequence of latent representations for network parameter θ.

Glow-TTS utilizes two-loss criteria, duration loss, and log-likelihood loss, as described in Equation 2.36. During the training phase, the duration predictor is trained to estimate the duration labels estimated by monotonic alignment search A, as a mean square error in the logarithmic domain, described by Equation 2.37. In the inference phase, statistics of the conditional prior distribution $P_Z(z|c)$ are predicted from the output of the text encoder and duration predictor. Afterward, sampled latent variables from conditional prior distribution are passed through Glow decoder to transform latent variables to Mel spectrogram.

$$\mathcal{L}oss = \mathcal{L}_{duration} + \mathcal{L}_{log-likelihood} \tag{2.36}$$

$$\mathcal{L}_{duration} = MSE(DP(sg[\mu_{1:T_{text}}]), d_{optimal}) \tag{2.37}$$

Glow-TTS heavily relies on Glow decoder, which follows the same architecture of affine coupling detailed in WaveGlow neural vocoder [Prenger et al., 2019] without local conditioning. Glow decoder comprises activation normalization, 1x1 invertible convolutional layer, and affine coupling, detailed in Section 2.3.2. Glow-TTS is trained using maximum likelihood estimation along with mean squared error loss on duration labels. The obtained results on Glow-TTS showed its ability to synthesize speech 15.7 times faster than Tacotron 2 with matching performance in terms of mean opinion scores [Kim et al., 2020].

2.8.3 Grad-TTS

Grad-TTS is a non-autoregressive approach based on a diffusion probabilistic model used as a decoder network. As illustrated in Figure 2.11, Grad-TTS comprises a text encoder, duration predictor, monotonic alignment search, and decoder network. The components of Grad-TTS such as text encoder, duration predictor, and monotonic alignment search are implemented in the same way as in Glow-TTS, except for the decoder network. As stated in Section 2.3.3, DPM principally consists of the forward and reverse diffusion processes. In the forward diffusion process, output data is deconstructed till some simple distribution like Gaussian distribution with zero mean and unit variance is achieved. Conversely, the reverse diffusion process works towards constructing the output distribution and learning the trajectories of the reverse-time forward diffusion by a parameterized neural network, s_θ in the DPM framework [Nichol and Dhariwal, 2021, Ho et al., 2020, Popov et al., 2021]. The DPM decoder network based on s_θ is implemented using U-net architecture as designed in [Ronneberger et al., 2015].

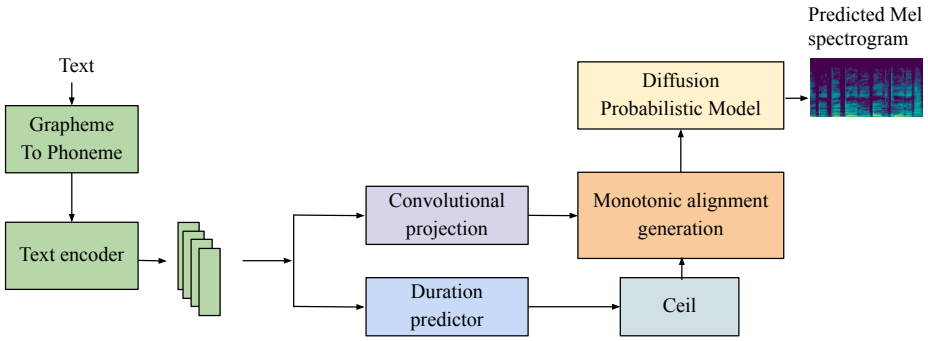

Figure 2.11 – Block diagram of the Grad-TTS system architecture

The text sequence, $c_{1:T_{text}}$ is given to text encoder network for generating text encoder output, $\tilde{\mu}_{1:T_{text}}$, where T_{text} is length of text sequence. Furthermore, text encoder output $\tilde{\mu}_{1:T_{text}}$, is provided to alignment A to create a sequence of latent representation $\mu_{1:T_{Mel}}$, where T_{Mel} is the length of output Mel spectrograms. Grad-TTS optimizes the maximum likelihood of monotonic alignment search A and reduces MSE between phoneme duration estimated by A and output of duration predictor. The output aligned encoder output, $\mu_{1:T_{Mel}}$ is then passed to a decoder that transforms Gaussian noise parameterized by encoder output into Mel spectrogram.

The loss criterion to train Grad-TTS is the sum of three losses, namely $\mathcal{L}_{duration}$, duration loss, same as explained in Glow-TTS, $\mathcal{L}_{enc}$, encoder loss, and $\mathcal{L}_{diffusion}$ diffusion loss. The encoder loss term is incorporated to minimize the negative log-likelihood so that encoder output $\tilde{\mu}_{1:T_{text}}$ follows the normal distribution $\mathcal{N}(\tilde{\mu}, I)$. After that, the decoder receives the input noise parameterized using encoder output μ. Initializing input noise close to parameterized normal distribution plays a crucial role in learning the alignment and synthesizing realistic speech. During the training, A alignment is searched with the fixed encoder parameters. After fixing the alignment A, network parameters are updated in optimization to reduce the loss criteria. Finally, $\mathcal{L}_{diffusion}$ is estimated as weighted loss indicating gradients of log-density at different time steps. The diffusion loss term is based on the fisher divergence term described by Equation 2.29, and a detailed explanation is provided in Section 2.3.3. Diffusion loss enables the computation of change in noise density in each diffusion step parameterized using s_θ as a U-net-based decoder network.

As stated in Equation 2.37, the duration loss is the same as the one used in the Glow-TTS. The encoder loss, $\mathcal{L}_{encoder}$ is based on negative log-likelihood. Since the DPM framework starts decoding random noise with a normal distribution $\mathcal{N}(\mu_{enc}, I)$, it is easier for the DPM decoder to sample noise that is already close to the output distribution. As a result, it is advised to include $\mathcal{L}_{encoder}$ as one of the auxiliary loss terms. In [Popov et al., 2021] diffusion loss $\mathcal{L}_{diffusion}$ is described as the expectation of weighted losses associated with estimating gradients of log-density of noisy data at different time horizons, T, starting from $t = 0$. The loss criteria for Grad-TTS is stated as below in Equation 2.38,

$$\mathcal{L}oss = \mathcal{L}_{duration} + \mathcal{L}_{encoder} + \mathcal{L}_{diffusion} \tag{2.38}$$

Using a stochastic Oridnary differential equations (ODE) network allows using a flexible inference scheme by varying the number of steps required to estimate the trajectories of reverse diffusions. Therefore, it provides a trade-off between inference speech and the quality of synthesized speech. Grad-TTS has shown approximately twice faster as Tacotron 2 with comparably similar speech quality [Popov et al., 2021].

2.9 Expressive TTS

Expressivity is an inherent part of naturally produced speech in human interactions. The naturalness of synthesized speech relies on the expressiveness of the text-to-speech system. It is factored into various aspects of speech characteristics such as emotion, prosody, textual content, timbre, speaking style, etc. [Tan et al., 2021].

In the context of expressive speech synthesis, for an input text, speech can be synthesized in various expressivity classes. Thus it is imperative to learn the one-to-many mapping corresponding to possible variations in synthesized speech for a single text.

The possible variations in speech refer to suprasegmental features such as pitch, energy, timbre, aperiodicity [Tan et al., 2021]. Therefore, modeling expressive TTS without explicit underlying information of expressivity will result in predicting the average Mel spectrogram instead of understanding the underlying expressivity characteristics [Toda and Tokuda, 2005, Takamichi et al., 2016]. In 2009, Schröder proposed to categorize the expressive speech synthesis approaches into three main classes; namely, expressive speech synthesis using explicit control, playback approach, and implicit control [Schröder, 2009]. Now, we will overview various approaches proposed to develop an expressive TTS system with various speech synthesis frameworks

2.9.1 Formant Synthesis

The first few efforts toward building expressive speech synthesis mainly focused on extending the formant speech synthesizer by adjusting acoustic and prosodic features. Formant expressive speech synthesis systems proposed in [Cahn, 1990, Murray and Arnott, 1993, Burkhardt and Sendlmeier, 2000] explored various rule-based approaches to fine-tune the acoustic features. The Affect editor developed by Cahn was the first attempt to synthesize emotional speech using a formant synthesizer [Cahn, 1990, Schröder, 2001]. For example, for anger [Murray and Arnott, 1995] expressivity class F0 is set to 10 Hz higher, range +9 HZ, and loudness to +6db.

Based on a commercial formant system, DECtalk, Murray, and Arnott presented an expressive speech synthesis system HAMLET [Murray and Arnott, 1993]. HAMLET system synthesizes expressive speech by modification in pitch and durations based on rules extracted for each expressivity class. The perceptual listening experimentation suggested that mean pitch, pitch range, speech rate, phonation, and vowel precision play a crucial role in expressivity control in formant synthesizers [Williams and Stevens, 1972, Fairbanks and Pronovost, 1939b]. The importance of pitch and duration in expressive speech synthesis control was investigated in various works [Schröder, 2009, Montero et al., 1999].

2.9.2 Unit Selection

The common approach in unit selection speech synthesis is to concatenate speech segments corresponding to the sequences of sounds of given text to synthesize. After that, the Pitch Synchronous Time Scaling (PSTS) algorithm is employed to modify acoustic excitation and prosodic features. The modified prosodic features include pitch, duration, and intensity, while the excitation parameters were jitter, shimmer, and glottal wave. The presented approach in [Vroomen et al., 1993] investigated diphone expressive speech synthesis for seven expressivity classes neutral, joy, boredom, anger, sadness, fear, and indignation. The obtained results highlighted that intonation and duration are sufficient to synthesize expressive speech using a diphone synthesizer. The rule-based method constrains explicit control over expressivity, which makes it difficult to synthesize expressive

speech in a natural human voice. The set of rules is determined by the quality of the data analysis, the rules, the number of control factors, and the literature review results.

The playback approach for expressive speech synthesis is either playing back from an existing expressive speech database or using a system trained to synthesize a particular class's expressive speech. The playback approach for unit selection and HMM-based expressive synthesis systems includes the work presented in [Yamagishi et al., 2003, Hofer et al., 2005, Fernandez and Ramabhadran, 2007]. In 2000 Iida et al [Iida et al., 2000] proposed the storage of large databases for each emotion. The respective expressive speech dataset is used to select units for synthesis in target expressivity at the inference time. In the playback approach, the expressive speech is synthesized independently using the respective expressive speech dataset. Here, expressive speech synthesis is implemented either by merely playing back what is available in the database of the target expression or using the models trained using the target expression database.

In [Hofer et al., 2005] expressive datasets with expressivity class of angry, happy, and neutral are unified for synthesizing expressive speech using the unit selection approach. The cost function of units is structured to provide more penalties to units other than the target expressivity class. Similarly, [Fernandez and Ramabhadran, 2007] proposed the use of combining various units of expressivity classes to synthesize expressive speech. Due to dependence on the existing expressive speech dataset, the playback approach faces difficulty in scaling expressive synthesis to a wide range of application domains.

2.9.3 Statistical Parametric Synthesis

The interpolation technique provides flexible control over expressive speech synthesis systems based on a data-driven approach. In 2005, Turk, Schröder, Bozkurt, and Arslan proposed interpolation of the spectral envelope of neural speech and expressive speech to generate an estimate of expressive speech [Türk et al., 2005]. The method can be incorporated into unit selection [Schröder, 2007], allowing for the continuous interpolation between a source and a target synthesized speech.

The principle of interpolation is the same whether the target voice is recorded as in the playback approaches described above or whether it is the result of a mapping process in a voice conversion system [Govind and Prasanna, 2013]. Statistical parametric TTS systems based on hidden Markov models have widely been used for expressive speech synthesis. Additionally, expressivity transfer is also performed using averaging the style model to the desired style [Masuko et al., 2004]. Style control code for the desired style is opted at the inference time to transfer expressivity in the neutral statistical parametric TTS system. Using interpolation or transfer, the averaging model results in over smoothing of acoustic features estimated by the TTS system [Barra-Chicote et al., 2010]. Thus it affects the naturalness of synthesized expressive speech. Multiple regression hidden semi-Markov model [Nose and Kobayashi, 2013] has also been presented to transfer the expressivity using style codes.

2.9.4 Neural Network Synthesis

In [An et al., 2017], expressive speech synthesis was developed by two methodologies, first by retraining a neutral neural network model and adding emotion codes to each layer of the neural network model. The detailed study was conducted in [Inoue et al., 2017], to understand the control of variations in speaker and expressivity in the neural network approach. Although similar to speaker adaptation, emotional adaptation is not trivial because emotional speech has strong prosody variations that are difficult to model [Besson et al., 2002, Paeschke, 2004, Hashizawa et al., 2004, Hashizawa et al., 2004].

In [Ohtani et al., 2015], expressivity transfer was accomplished by parameters of an eigen-voice space of neutral speech and the differences between neutral and expressive speech. In [Chen et al., 2015], two distinct cluster adaptive training (CAT) subspaces were created to create representations of speaker and expressivity. For expressivity transfer, the expressive acoustic features are projected into the CAT subspace of the expressivity class to estimate the expressive CAT weights. After that, the neutral speech is then projected into the subspace of the target speaker to obtain the speaker CAT weights. In [Xue et al., 2018] three architectures were proposed for developing RNN-LSTM based parametric TTS system for expressivity adaptation, namely emotion-specific retraining of TTS for each emotion, emotion code as an additional input to a single TTS system, and emotion code provided to each hidden LSTM layer.

2.10 Controlling And Expressivity Transfer

This section discusses various approaches proposed for developing end-to-end TTS systems to transfer the expressivity. On the basis of the scale at which expressivity code control is provided as an additional input to the end-to-end TTS system, expressivity transfer approaches can be divided into two groups. First, coarse-grained expressivity transfer incorporates usage of time-invariant global expressive embedding provided as input to decoder network of end-to-end TTS systems along with text encoder output [Shah et al., 2021, Mohan et al., 2021, Karlapati et al., 2020, Zaïdi et al., 2021, Wang et al., 2018, Zhang et al., 2019, Aggarwal et al., 2020, Hsu et al., 2019] . Fine-grained expressivity transfer includes deriving local correlates between textual content and reference acoustic features to represent as expressivity embedding at various scales such as phoneme, utterance, and segmental [Yang et al., 2020a, Akuzawa et al., 2018, Karlapati et al., 2020, Sun et al., 2020b, Sun et al., 2020a].

As the expressivity transfer approaches are trained with the same reference Mel spectrogram as text content, expressivity embedding might encode the textual information. This phenomenon is called content leakage [Hu et al., 2020], which affects the intelligibility of synthesized speech. Information sieve architecture was proposed in [Dai et al., 2021], in which a sieve layer was presented to filter out redundant information.

The work done in expressivity transfer is mainly focused on expressivity as a prosodic variation from input reference acoustic feature to transfer in desired speaker's synthesized speech. Most end-to-end TTS systems presented the results on Blizzard challenge 2013 [Watts et al., 2013], LibriTTS [Zen et al., 2019], and VCTK [Veaux et al., 2017], blizzard

2019 speech datasets, in which no explicit labels were available as an expressivity class.

Over the years, a limited study has been conducted to transfer expressivity for each emotion class rather than expressivity as a reference prosodic representation. Recent deep learning models illustrated significant performance improvement in expressivity transfer. It is still limited to parallel transfer, where the same acoustic features are provided as textual content. Non-parallel expressivity transfer involves providing mismatched reference acoustic features and textual content. The results obtained with various approaches showed degradation in overall expressivity transfer performance. Dependency on textual content may create bottleneck scenarios to require appropriate reference acoustic features in desired expressivity. Therefore, expressivity transfer approaches need to address various issues such as speaker leakage, content leakage, and non-parallel expressivity transfer.

Coarse-grained Expressivity Transfer

In [Skerry-Ryan et al., 2018] reference encoder was proposed to map variable-length reference Mel spectrogram into fixed-length expressivity representation. The reference encoder is composed of convolutional layers with batch normalization and ReLU activation functions. Global style token (GST) was proposed as an extension to the previously mentioned approach trained along tacotron system [Wang et al., 2018]. Global style token relies on coarse-grained expressivity representation. In the GST approach output of reference, the encoder was passed through multi-head attention to learn the stylistic factors from the reference Mel spectrogram. Various approaches have been proposed to use VAE based expressivity encoder to create representation given to decoder network of end-to-end TTS [Akuzawa et al., 2018, Aggarwal et al., 2020, Zhang et al., 2019]. In [Akuzawa et al., 2018] VAE framework is trained along with a voice-loop decoder to learn the speech expressions as a tractable distribution. Hence, VAE based expressivity encoder provides a flexible framework for interpolation and semi-supervised learning. VAE based expressivity encoder was presented as an augmentation to the Tacotron 2 framework, where the output of the reference encoder was used to create a latent representation of expressivity [Zhang et al., 2019].

The disentanglement of expressivity representation makes it easier to generate a clustered representation of expressivity classes, allowing for flexible control. Hierarchical latent representation approach based on Gaussian mixture VAE [Dilokthanakul et al., 2016] was presented in [Hsu et al., 2019] to scale representation at two levels. First, attribute groups as expressivity and second level conditioned on the previous latent variable as a multivariate Gaussian variable. The second level latent variable features factored control over noise level and speaking rate of a given attribute group. The GMVAE as expressivity encoder was trained jointly with Tacotron 2 framework similar to VAE [Zhang et al., 2019] and GST [Wang et al., 2018]. An extension to DCTTS [Tachibana et al., 2018] was proposed in [Tits et al., 2021] along with VAE for controlling the seven emotions. The reference encoders were based on LSTM based discriminative model pretrained on expressivity recognition.

In [Aggarwal et al., 2020], Householder normalizing Flow [Tomczak and Welling, 2016] was used as a way to perform the variational inference through the output of the reference encoder in the Tacotron 2 framework. The proposed approach improves the KL divergence

term and the reconstruction performance. Thus, Householder Normalizing Flow provides a more tractable posterior distribution estimate of expressivity.

STYLER, a non-autoregressive TTS framework, was proposed by [Lee et al., 2021] by modeling various style factors such as duration, pitch, energy, and noise. Along with text encoder output, various other latent representations are derived for each style factor. Hence, the length regulator conditions these latent representations to generate pitch embedding, energy embedding concatenated together as an input to the decoder network. Recently proposed STYLER adapts domain adversarial training for robust expressivity transfer. A universal multi-speaker, multi-style TTS model, was proposed by [Paul et al., 2021] which aims at minimizing the Renyi divergence between content style and speaker style using an adversarial training framework. Adversarial framework for training expressive TTS system enables the flexible and robust expressivity transfer. In [Li et al., 2021a] provides an explicit constraint on emotion embedding and speaker embedding by emotion match loss trained along with Tacotron 2 as the backbone TTS model. The proposed work illustrates the results of six emotion classes stated in Ekman's theory of emotions. Recently reinforcement learning-based training framework was proposed in [Liu et al., 2021] for training Tacotron-GST as an agent and speech emotion recognition as a reward function.

Fine-grained Expressivity Transfer

Even though coarse-grained approaches have shown promising results, they still lack fine-grained control over expressivity transfer, which considers phoneme level information. Few research works have been focused on the fine-grained transfer of prosody, which includes prosody of speaking style either using LibriTTS corpus or audiobook corpus [Sun et al., 2020a, Sun et al., 2020b, Klimkov et al., 2019]. A fine-grained VAE framework is proposed to extract latent variables at each token of phoneme embedding [Sun et al., 2020a]. This approach uses sequential prior in a discrete latent space implemented with the help of vector quantization, where each token of phoneme embedding is aligned to the target Mel spectrogram. This work is further extended by creating multi-level alignment for phoneme, word, and utterance [Sun et al., 2020b].

During the expressivity transfer, one of the challenges is to tackle the source speaker leakage, where synthesized expressive speech has speaker identity from the source speaker of reference Mel spectrogram. The CopyCat architecture was proposed to reduce the effect of speaker leakage by conditioning the decoder on upsampled phoneme encoder output, speaker embeddings [Karlapati et al., 2020]. The CopyCat architecture uses GAN [Goodfellow et al., 2014] based discriminator to reduce the speaker leakage. In CopyCat architecture, the output of the phoneme encoder is first upsampled. Subsequently, it is given to the expressivity encoder. As a result, CopyCat architecture's expressivity representation includes fine-grained control over expressivity transfer.

Chapter 3

Datasets and Preprocessing

3.1 Introduction

In 1957 a US Army Specialist, William D. Mellin stated that computers cannot think for themselves, and that "sloppily programmed" inputs inevitably lead to incorrect outputs [Darryl, 2018]. Furthermore, according to the garbage in, garbage out principle, flawed, or nonsense input data produces nonsense output or "garbage". The machine learning techniques are continuously improving just within a few years, but the data used for training such models play a vital role in the machine learning frameworks. Deep learning models understand the complex latent structures in the input data distribution to produce the desired output. The quality of the dataset with which machine learning models are trained highly influences the performance of the system. Thus data preparation is a crucial stage in the development of machine learning-based systems. In this chapter, we will overview the data preparation in the context of the speech synthesis framework.

Text-To-Speech (TTS) systems have to learn the underlying transformation function to produce the speech waveform for given input text. But to train such systems with deep learning techniques requires preprocessing to create a representation that eases the learning process for the deep learning models. In this chapter, we will present the speech datasets, which include the speech datasets for English as well as for French. After that, we will describe the textual preprocessing used for developing a parametric TTS system and for developing an end-to-end TTS system. The textual preprocessing maps raw text into input features, which are presented as input to the TTS system. Then, we will present the parametrization of speech waveforms with signal processing based WORLD vocoder and neural vocoder.

3.2 Speech Synthesis Datasets

The design and recording of datasets for developing TTS systems encounter several problems, such as the collection and selection of texts for target application domains, and the selection of suitable speakers. Furthermore, the technical skillsets for insuring and maintaining the quality of recordings are also essential to analyze the erroneous

pauses, pronunciation errors and ensuring uniform acoustic conditions throughout the recording of the dataset [Oliveira et al., 2008]. Additional precautions are advised for recording expressive speech datasets, such as quality of the enacted emotions, consistency in recordings of emotions, and the strength of each emotion.

Expressive speech synthesis datasets are prepared in two ways. Expressive speech recorded by actors [Burkhardt et al., 2005, Pitrelli et al., 2006, Fairbanks and Pronovost, 1939a, Whiteside, 1998], and the second spontaneous expressive collected from real-life situations such as [Williams and Stevens, 1972, Johnstone and Scherer, 1999, Campbell, 2005, Ishi and Campbell, 2004]. We focused on the expressive speech datasets developed with the help of actors.

A speech dataset is referred to as a speech and the corresponding text. Before the usage of speech dataset, various steps are to be followed to use the recorded speech dataset for a TTS system such as re-sampling of speech, normalization etc. After the preparation of the speech dataset, a human intervened manual analysis is performed to evaluate overall quality of recorded speech. For audio data, the overall quality of audio signals is reviewed in terms of acoustically unclean data for instance noise, background noise, and acoustic artifacts. With this subjective analysis, the unclean audio samples are removed from the corpus. This is one of the crucial steps, if unclean noisy audio samples are present in the corpus, it might lead to poorly synthesized speech or poorly trained acoustic models of the TTS system. In addition to the quality of audio data, suitable sampling rate, encoding type, and bit rate have opted across all the audio data. Similarly, text analysis is conducted to observe the text encoding, spelling mistakes, formatting marks, and if the textual content corresponds to the associated speech utterance of a given sample.

In this section, we will overview the speech synthesis corpora for English and French languages. We have used the presented corpora for building an expressive TTS system based on both parametric TTS and end-to-end TTS. We termed expressivity classes to denote various emotions and neutral voice present in speech utterances.

EmoVDB dataset

Emotional Voice Database [2] (EmoVDB), an open-sourced emotional speech synthesis dataset, was developed for an English expressive TTS system [Adigwe et al., 2018]. The EmoVDB dataset is multispeaker emotional speech recorded from two female speakers and two male speakers. EmoVDB dataset incorporates five expressivity classes with four emotions and one neutral. The expressivity classes used in the EmoVDB dataset are amusement, anger, sleepiness, disgust, and neutral. The expressivity classes are selected to cover the diverse emotional space described by the Russel circumplex [Russell, 1980]. Therefore, interpolation techniques can be employed to synthesize speech for expressivity classes not present in the dataset.

The dataset is recorded in anechoic chambers at the Northeastern University and the University of Mons. All the speech data were recorded at a 44.1k Hz sampling rate. In the scope of thesis work, we downsampled the speech data at 16k Hz stored in 16-bit PCM encoding format. During the recording sessions, 5-10 min of a break was given between

2. https://github.com/numediart/EmoV-DB

Table 3.1 – Total duration for 4 speakers and 5 expressivity classes in EmoVDB dataset

Speaker	Gender	neutral	amused	angry	sleepy	disgust
Speaker 1	Female	56 min	29 min	68 min	61 min	25 min
Speaker 2	Female	48 min	40 min	41 min	68min	45 min
Speaker 3	Male	64 min	65 min	61 min	64min	64 min
Speaker 4	Male	39 min	39 min	-	34 min	-

each 30 min of recording. All the recording sessions lasted for several days to incorporate the availability of actors. The mispronounced sentences were rerecorded by actors. The subset of textual transcripts of the CMUarctic dataset was used for the recording of the EmoVDB dataset. Additionally, the transcripts of the CMUArctic dataset were annotated at utterance and phonetic levels. The speech waveforms and text were aligned using forced alignment systems presented in Train_align technique [Brognaux et al., 2012].

LJ Speech dataset

The LJ Speech [3] (LJS) dataset was designed for speech synthesis systems as a public domain speech dataset, which was recorded by the Librivox project [Ito and Johnson, 2017]. LibriVox project aims at creating free public domain audiobooks with the help of worldwide volunteers.

In total, 13,100 speech utterances were recorded by a single female speaker in the English language. These audio clips are from passages of non-fiction books with corresponding transcriptions. The texts were published between 1884 and 1964 and are in the public domain. The speech utterances were recorded for short lengths from 1 to 10 sec, contributing to approximately 24 hrs. All the audio clips were stored at a 22050 Hz sampling rate and stored in 16-bit PCM encoding format.

LibriTTS dataset

LibriTTS [4] is a multi-speaker English speech dataset recorded at a 24k Hz sampling rate, prepared by Heiga Zen with the assistance of Google Speech and Google Brain team members [Zen et al., 2019]. LibriTTS dataset is designed for TTS research. To our knowledge, LibriTTS is the largest TTS target dataset containing approximately 585 hrs of speech. This dataset is extracted from the audio and textual data taken from the LibriSpeech corpus [Panayotov et al., 2015]. The LibriSpeech corpus was prepared by automatically aligning audiobooks and their texts, segmenting them into utterances, and filtering noisy transcripts and audio recordings.

We used the train-clean-100 subset of the LibriTTS dataset covering 123 speakers and 25 mins on average per speaker. We excluded the speakers for which speech utterances have a duration of less than 5 min and more than 10 sec.

3. https://keithito.com/LJ-Speech-Dataset/
4. https://openslr.org/60/

Canadian French emotional dataset

The Canadian French emotional [5] (CaFe) speech dataset was developed for the study of emotions. The CaFE dataset is the first Canadian French emotional dataset, which contains a total of 936 speech utterances [Gournay et al., 2018]. This dataset consists of six different sentences recorded by six male and six female actors. The dataset adapted Ekman's theory to define the six basic emotions, sadness, happiness, anger, fear, disgust, and surprise [Ekman, 1999]. Each of the emotions acted in two different intensities (low and strong). All the selected actors were native Canadian French speakers with a Québec French accent. The recording sessions took place, one actor, at a time, in a professional soundproof room. The speech utterances were recorded at a high-resolution 192 kHz sampling rate, 24 bits per sample.

Caroline dataset

We used the French Caroline audio-visual expressive speech dataset for conducting expressivity transfer to target speaker's voice. Caroline dataset was designed for audio-visual expressive speech synthesis for French language [Dahmani et al., 2019]. The Caroline audio-visual dataset is developed internally in the Multispeech. Caroline dataset consists of 2000 sentences recorded by a semi-professional actress in neutral mode. After that, six emotions, namely joy, sadness, fear, surprise, disgust, and anger, are recorded as a subset of the Caroline dataset. For each subset, 500 sentences are used for building expressive speech datasets. The neutral dataset consists of approximately 3hrs of recordings, while the emotional subset of the Caroline dataset consists of 1 hr of recordings. All the speech signals are recorded at 16k Hz.

Before recording the Caroline dataset, a textual transcript with maximum phonetic coverage was created from the French textual datasets. Initially, 7000 French textual transcripts were collected from the internal-team datasets and the open-source datasets. Then, a greedy algorithm is applied to extract 2000 sentences for which the best coverage rate is obtained on phoneme bigrams (diphones). The neutral Caroline dataset covers approximately 92% of the French diphones, and the emotional subsets cover 52% for each emotion. In total, both neutral and emotional datasets contribute to 100% coverage of French phonemes.

In Caroline expressive dataset, each subset was recorded in one session for each emotion [Dahmani et al., 2019]. During the recording, the session actress is provided with a description of each emotion. Also, the set of scenarios are provided to the actress as additional information. These scenarios assisted in emulating the acting technique of Stanislavski [Moore, 1984]. The Stanislavski technique allows the actress to dig into her own affects to create an emotion, taking advantage of her emotional memory. The method is often shown as particularly naturalist, as opposed to a more figurative performance [Stanislavski and Vilar, 1984]. The actress was presented with three scenarios for each emotion. The actress picked up the scenario that felt the closest to her affective experience. The emotional scenarios (in French) given to the actress were taken from the GEMEP (GEneva Multimodal Emotion Portrayals) dataset [T. Bänziger and Scherer, 2012].

5. https://doi.org/10.5281/zenodo.1219621

Table 3.2 – Total duration of expressivity classes in Caroline dataset

Expressivity classes	Duration	Subset %
neutral	3 hr 58 min	35.86
joy	57 min	8.7
sadness	1 hr 7 min	10.1
fear	1 hr 10 min	10.65
surprise	1 hr 4 min	9.6
disgust	1 hr 51 min	16.85
anger	55 min	8.21

We used the Caroline dataset for developing various techniques to transfer the expressivity in the TTS system. Although the Caroline dataset contains additional visual expressive data, we only considered audio information for synthesis.

Synpaflex dataset

The Synpaflex[6] dataset is a French expressive speech synthesis dataset built for prosodic models in the framework of storytelling [Sini et al., 2018]. This dataset is built as a part of the Synpaflex research project focusing on improving expressivity in the text-to-speech synthesis framework. The main objective of the Synpaflex project is to develop an audiobook dataset with a large amount of speech, containing different linguistic, acoustic, phonetic, and phonological phenomena recorded by a single female French speaker.

The Synpaflex corpus consists of 87 hrs of audiobooks collected from the Librivox website, read by a single speaker covering various types of literature namely novels, short stories, tales, fables, and poems. The inherent speaker's expressive speaking style and personification of characters are distinguishable during the reading for audiobooks. This enabled to study long-term discourse strategies incorporated by a speaker. For aligning large text with audio-book speech, a French speech-to-text aligner was used to conduct the segmentation at the phone level. The broad phonetic transcription, based on the French subset of Sampa, has been extracted and aligned with the speech signal using JTrans [Cerisara et al., 2009]. Before the alignment step, collected text from various sources was unified for formatting and encoding purposes. Thereafter, the text is normalized and split into paragraphs using rule-based techniques suitable for the French language. Then, each chapter was processed separately to segregate linguistic information.

In the Synpaflex corpus, speech segments were annotated manually for prosodic units, characters, emotions, and other events from the audiobooks. In total 6 emotional classes were used in the emotional annotation process. These six emotions are based on Basic Emotions theory [Ekman, 1999], namely sadness, anger, fear, happiness, surprise, disgust. Additional two emotional classes were selected to represent the content of different books, which are irony and threat. For each emotion, intensity levels on a scale of 1 to 3 were used to assign the emotional intensity present in the spoken sentence. For instance, intensity

6. https://synpaflex.irisa.fr/

Table 3.3 – Total duration of expressivity classes in Synpaflex dataset

Expressivity class	Duration	Subset %
neutral	8 hr 11 min	61.0
joy	32 min	3.9
sadness	44 min	5.4
fear	11 min	1.3
surprise	4 hr 42 min	35.0
disgust	15 min	1.9
anger	31 min	3.9
irony	10 min	1.2
threat	3 min	0.4

slightly angry, 1, angry 2, and strongly angry 3. All the speech signals are stored at a 44.1k Hz sampling rate.

In the Synpaflex corpus, expressive speech samples are available but due to an insufficient number of speech samples for each emotion as well as the unbalanced distribution of emotional speech samples, we have used only the neutral voice of Synpaflex in our work. For instance, only 15 min recordings of disgust emotions are available in the Synpaflex dataset with three different subclasses indicated by emotional intensity level. This makes it difficult to use the Synpaflex expressive dataset consistent with Caroline expressive dataset.

Lisa dataset

The Lisa speech synthesis dataset was developed internally in the Multispeech team, Loria laboratory. The Lisa dataset was initially built for neutral speech synthesis by concatenation and was particularly used in Soja TTS System. The dataset is composed of 1812 sentences extracted from "Le Monde" newspapers. These sentences have been selected between four million French sentences by a greedy algorithm to cover as best as possible a set of criteria. The criteria were based on the diphone, syllable, word, position in the sentence, and rhythmic group information. The developed algorithm selected sentences to obtain all diphones of the French language in several positions (beginning - end of word, between two words, beginning-end of rhythm group, begin-end of sentences, none of these positions). All these positions are representative of prosody in the French language. The greedy algorithm minimized the number of sentences, and all kinds of sentence lengths are present. The corpus has been recorded by a female voice in several sessions in a calm room (with wall treatment). It lasts around 3hrs and has been recorded at 16k Hz.

Table 3.4 – Total duration of reading style in Siwis dataset

Reading style	Duration	Subset %
parl	4 hr 02 min	29.3
book	5 hr	36.3
siwis	3 hr 08 min	22.7
sus	1 hr 26 min	10.4
emph	11 min	1.2

Simple4All Tundra dataset

The Simple4All Tundra [7] (Tundra) is a multi-lingual corpus designed for TTS research in audiobooks [Stan et al., 2013]. The Tundra dataset aimed to create TTS systems for multiple languages. The corpus incorporates approximately 60 hr of speech data collected from audiobooks in 14 languages. The Tundra corpus was developed with minimal language-specific knowledge and manual intervention. In addition to text and speech alignment, speech denoising and dereverberation algorithms have been applied as a post-processing step.

We used the subset of the Tundra corpus corresponding to the French language, recorded in a neutral voice. The French audiobook consists of approximately 90mins of speech data for which corresponding textual content was available at the sentence level. All the speech signals are stored at 16k Hz.

Siwis dataset

The Siwis speech synthesis dataset was designed for building French TTS systems [Honnet et al., 2017]. The dataset also investigated multiple styles and emphasis recorded in speech data. The Siwis dataset contains more than 10hrs speech and is freely available for public use. The speech was recorded by a French Female voice actress, which also included emphasized words in various contexts. The professional voice recording agency, Voice Bunny, conducted the recording of the Siwis dataset. Instructions were presented to the native French female speaker, such as emphasizing a specific word in a sentence, reading the chapter with emotions, and long pauses in between sentences.

The data was recorded using Studio Projects B1 microphone and provided in 44.1k Hz mono-channel 16-bit PCM encoding format. Adobe Audition 9 was used for processing. Thirty minutes of recordings generally required 90 minutes of processing, editing, and checking from the voice actress. The silences and long pauses were obtained using automatic alignment performed between phonemes and the speech segments. The phoneme information is derived from contextual labels created using the eLite12 web service.

Siwis dataset incorporates five reading styles parl, book, Siwis, sus neutrally read the sentence as per the source of the text, while emph, emphasis, and a chap with expressive style. In the context of the doctoral thesis, we only used parl, book, siwis, sus styles to

7. https://simple4all.org/

incorporate a neutral manner of voice.

3.3 Text Preprocessing

For a demonstration of the transfer of expressivity, we experimented with two TTS frameworks, parametric TTS and end-to-end TTS. In a parametric speech synthesis approach, the first step consists in converting raw text into a representation that will convey linguistic, phonetic, and prosodic information, resulting in a sequence of contextual labels (also called contextual features) [King, 2011]. Each item of the sequence corresponds to a phone or a silence segment. Contextual labels for a given speech segment (whether a phone or silence segment) correspond to a complex tree-structured representation which is flattened at phoneme, syllable, and word level with prosodic and phonetic information. This information is further enhanced with context labels which provide information about adjacent phonemes, positions of adjacent and current phonemes, stress and accent information about these phonemes, part of speech for a word to which the phoneme belongs to, etc.

This format is commonly used in HMM-based speech synthesis tool (HTS) and is also known as HTS format labels [Zen, 2006]. This format is widely used in parametric speech synthesis systems. We used Soja, a speech synthesis tool developed by the Multispeech team at LORIA laboratory, Nancy, as a front end for text processing and label generation. Furthermore, after the generation of contextual labels, the corresponding input vector is created with the aid of the question file. The question file is a simple set of regular expression questions which has binary values (1 when regular expression matches the formatted label, 0 otherwise). These questions are prepared considering the characteristics of the language. Examples of questions (which are then formalized as regular expressions) are: is the current phoneme a vowel or not? is the current phoneme a nasal or not? is the current phoneme is a fricative or not? is the current phoneme stressed or not? etc. In our experiments, the question file for the French language is designed with 180 questions that consider pentaphone information, with 36 unique phone sets.

After generating contextual labels for the text, speech waveforms and contextual labels are aligned at the phoneme level with the Hidden Markov Model based forced alignment tool by HTK toolkit [Young, 1994]. This forced alignment relies on Mel Frequency Cepstral Coefficient (MFCC) acoustic features, and provides the beginning and ending time of each phone segment. The resulting timing information will later be used to train the duration model, and; the duration model will be used to predict the duration of each phoneme during the synthesis step. As a common practice in deep learning, we normalized the contextual feature vector created from the 180 questions for smooth convergence of neural network parameters. The features were normalized using min-max normalization to a range of [0.01,0.99].

The explicit textual preprocessing is not required for the end-to-end TTS framework apart from grapheme to phoneme conversion. We used the Soja tool for converting text to a sequence of phonemes. We assigned integer ids to each phoneme in the range of 1 to 36 and 37 for representing word breaks.

3.4 Speech Preprocessing

In parametric speech synthesis, speech waveform is represented using acoustic features. The essential requirement for parametrization of speech waveform is acoustic feature should allow reconstruction of the speech waveform using a vocoder. Therefore, parametric speech synthesis aims to predict the acoustic features of speech waveform and provide it as input to the vocoder.

For parameterization of the speech signal, we used the WORLD vocoder, which allows for real-time processing, and synthesizes high quality speech [Morise et al., 2016]. In 2009, Masanori Morise developed the WORLD vocoder, which is open-source, and freely available [Morise et al., 2016]. It consists of three speech analyses: spectral envelope, band aperiodicity, and fundamental frequency (F0). Further details of the WORLD vocoder are presented in Section 2.7.1. We used the WORLD vocoder throughout the thesis work to compute 187 acoustic features for every 5ms time frame, namely 180 spectral features (60 Mel Generalized Cepstrum (MGC) with first and second derivatives), three frequency-related parameters, i.e., one log fundamental frequency (Lf0) with first and second derivative and three excitation parameters, i.e., one band-aperiodicity (bap) with first and second derivative and one value for voiced-unvoiced information (vuv). In the case of a single speaker, the parametric TTS system, mean (μ), and standard deviation (σ) is computed on a given speech dataset. On the other hand, the mean (μ) and standard deviation (σ) are estimated over multiple datasets to build a multispeaker parametric TTS system. Based on the mean value (μ) and standard deviation values (σ), the acoustic features extracted from the WORLD vocoder were z-normalized.

The Mel spectrogram is created by transforming the spectrogram's frequencies to the Mel scale, where the spectrogram is computed using Short Time Fourier Transform (STFT). The basic idea behind using the Mel scale is that not all frequency bands in the linear scale spectrogram are equally important in human speech. In-depth details regarding Mel scale, Mel spectrogram, and STFT are detailed in Section 2.5. We computed Mel-spectrograms with 80 bins using librosa [8] Mel filter defaults. We apply the STFT with an FFT size of 1024, hop size of 256, and window size of 1024 samples. We used librosa library for resampling of speech waveform to desired sampling rate [McFee et al., 2015]. No additional pre-emphasis filtering is applied to the waveform signal before spectrogram calculation.

We used Siwis, Synpaflex, Lisa, and Caroline speech synthesis datasets for developing autoregressive end-to-end TTS systems (described in Section 6.2). Except for the Synpaflex and Siwis datasets, other French speech synthesis datasets were recorded at a 16k Hz sampling rate. Hence, initial experimentation with autoregressive end-to-end TTS systems was conducted with a sampling rate of 16k Hz. We trained WaveGlow from scratch on all the French speech synthesis datasets for coarse-grained end-to-end TTS approaches, namely Caroline, Lisa, Siwis, Synpaflex, and Tundra. WaveGlow neural vocoder was trained with default model hyperparameters and trained till 800k iteration steps as presented in the original implementation [9]. For fine-grained autoregressive TTS systems,

8. https://librosa.org/
9. https://github.com/NVIDIA/waveglow

first, we resampled speech datasets to 22050 Hz and used the pretrained WaveGlow model provided in the official implementation of waveGlow. The pre-trained waveGlow model was trained using the LJS dataset with the default model configuration.

We used Caroline and Siwis dataset to build non-autoregressive TTS systems described in Section 6.3. Recently proposed Hi-Fi GAN vocoder illustrated improved performance compared to WaveGlow and Wavenet vocoder [Kong et al., 2020]. Therefore, we opted pre-trained Hi-Fi GAN model as a neural vocoder for non-autoregressive TTS systems. The pre-trained Hi-Fi GAN model trained on LJS dataset with default configuration provided in official implementation [10].

We combined all the speech datasets for English and French (mentioned in Section 3.2) for developing a speaker encoder and expressivity encoder. These encoders were used for measuring cosine distances to estimate speaker similarity score and expressivity similarity score. A detailed explanation of similarity scores is presented in Section 4.2.2.

3.5 Discussion

In this chapter, we enlisted speech synthesis datasets for French and English used throughout the thesis work. Due to inconsistent emotional classes between Caroline, Siwis, and Synpaflex datasets and the amount of expressive speech data available for each expressive style or emotion, we excluded expressive speech samples from both the Synpaflex and Siwis datasets. We used the grapheme to phoneme component of the Soja tool [11] to convert textual transcripts of all the French speech datasets without considering the phoneme transcriptions provided in speech synthesis datasets.

In the case of the parametric TTS framework, contextual labels are obtained using the Soja tool, and they are then converted to 180-dimensional features. These textual features are used as an input to the parametric TTS system to predict the 187 acoustic features represented as Mel Generalized Cepstrum, log fundamental frequency, aperiodicity, and voiced-unvoiced flag. During speech synthesis generation, these acoustic features are given to the WORLD vocoder to synthesize speech waveforms.

End-to-end TTS framework, a sequence of phoneme identity as integer is given as input to predict the Mel-spectrograms. Then, the WaveGlow neural vocoder converts these Mel spectrograms to speech waveforms. The WaveGlow neural vocoder was pretrained with French speech synthesis datasets at a 16k Hz sampling rate, and more details are provided in Section 2.7.2. Any defect in data preparation may lead to an irreversible effect on the quality of speech synthesis, as data preparation is a time-consuming process. Therefore, data preparation is an important step in building a high-quality speech synthesis system.

10. https://github.com/jik876/hifi-gan
11. https://raweb.inria.fr/rapportsactivite/RA2010/parole/uid60.html

Chapter 4

Evaluation of Expressive Text-to-Speech Synthesis

4.1 Introduction

The performance evaluation of machine learning systems is an essential part of project development. Although the data preparation and development of the machine learning model are vital stages of the project, evaluation metrics play an equally important role in measuring the performance of the machine learning models. The usage of various evaluation metrics helps to understand the underlying behavior of the system on unseen inputs. Instead of using a single evaluation metric, the usage of different metrics helps validate the quality and robustness of the model.

It is expected from a machine learning model to generalize the learning process on unseen data, but learning to memorize is not desired behavior. Until the 1990s, the most common practice to measure speech quality was conducting subjective listening tests [Wang et al., 1992, Rix et al., 2006]. The objective evaluation provides a rough estimate of the performance of the TTS system. Generally, both objective evaluation and subjective evaluation are used for selecting the TTS system. This chapter will overview the techniques used in the evaluation of TTS and the performance of expressivity transfer. First, we will introduce the objective evaluation and two newly introduced evaluation metrics for the performance of expressivity transfer. Then, subjective evaluation with details about listening tests, and experimental setup.

4.2 Objective Evaluation

The objective evaluation metrics should correlate with the results of subjective listening tests to replace the expensive time consuming listening tests. Perceptual auditory models are commonly used in designing evaluation metrics for speech quality. These objective measures quantify the perceived quality of distorted speech relative to an undistorted reference sample [Kondo, 2018]. Additionally, objective evaluation methods such as Bark spectral density (BSD) [Wang et al., 1992], perceptual speech quality measure (PSQM), measuring normalizing blocks (MNB), perceptual analysis measurement system

(PAMS), and perceptual evaluation of speech quality (PESQ) [Rix et al., 2001] are used in evaluating audio quality as well as the quality of service [Takahashi et al., 2006].

For TTS systems, objective evaluation metrics should accurately represent the perceptual similarity between the original speech and synthesized speech segments. The choice of evaluation metrics also depends on the acoustic features used for representing speech waveform. Firstly, we describe the classical objective evaluation metrics Mel cepstrum distortion (MCD), root mean squared error log fundamental frequency (Lf0 RMSE), and distortion of band aperiodicity (BAP).

The perceptual quality of the TTS system heavily relies on the selection of the model after the training TTS systems. The most commonly used training objective for TTS is mean squared error, which does not always correlate with human perception [Baby et al., 2020]. The usual approach for evaluating perceptual quality is to design a perceptual listening test to rate synthesized speech utterances manually [Polkosky and Lewis, 2003]. Intelligibility refers to the ease of understanding the textual content in speech utterances. Nevertheless, the design and setup of such subjective evaluation of multiple TTS models can be time-consuming and expensive.

There are no standard evaluation metrics to measure the performance of expressivity transfer or prosody transfer. Several metrics from the speech processing field are adapted to reflect the acoustic correlate of expressivity [Skerry-Ryan et al., 2018]. One significant challenge in expressivity transfer is retaining the speaker's voice while synthesizing expressive speech. In addition to this, the unavailability of reference expressive speech utterances in the target speaker's voice obstructs the usage of traditional TTS evaluation metrics. Therefore, we need objective evaluation metrics that are highly correlated to the perceived expressivity and the speaker's voice in synthesized speech. In [Jia et al., 2018], speaker similarity scores are computed using cosine similarity metrics between synthesized speech and reference speech embeddings. We used a cosine similarity score to evaluate the transfer of expressivity. Cosine similarity allows measuring the strength of expressivity and the speaker's voice for transferred expressivity in the target speaker's voice.

4.2.1 Text-to-Speech System

The performance of TTS systems is usually measured using three objective evaluation metrics, MCD, Lf0 RMSE, and BAP. The mentioned objective evaluation metrics aim at estimating the error value for an associated acoustic feature such as Mel-Generalized Cepstral feature (MGC), Log fundamental frequency (Lf0), and band aperiodicity (bap). These acoustic features are extracted from speech waveform using the WORLD vocoder [Morise et al., 2016]. We recomputed the acoustic features from synthesized speech waveform to measure the objective evaluation metrics.

The TTS system may predict acoustic features of different lengths than the reference acoustic features. Before measuring the evaluation metric, we used the dynamic time warping (DTW) algorithm [Muller, 2007] to obtain an alignment between the reference acoustic feature and predicted acoustic feature [Kearney et al., 2009].

Mel cepstral distortion

The Mel cepstral distortion (MCD) is commonly used in speech applications such as automatic speech recognition, speaker recognition, speech synthesis, etc. MCD is the Euclidean distance between two vectors that describe the global spectral characteristics. Here, we used MGC to describe the speech signal, defined as the inverse Fourier transform of the logarithmic spectrum calculated on a warped frequency scale. The MCD presents an approximate estimate of perceptual distinction between phonemes [Vasilijevic and Petrinović, 2011] as stated below,

$$MCD_K = \frac{1}{T}\sum_{t=0}^{T-1}\sqrt{\sum_{k=1}^{K}(c_{t,k} - c'_{t,k})^2} \tag{4.1}$$

where $c_{t,k}$, $c'_{t,k}$ are the k^{th} MGC coefficient of the t^{th} frame from the reference and predicted MGC acoustic feature. We sum the squared differences over the first K^{th} MGCs, skipping $c_{t,0}$ (overall energy) [Skerry-Ryan et al., 2018].

Band aperiodicity distortion

Aperiodicity is defined as the power ratio between the speech signal and the aperiodic component of the signal. Since this power ratio depends on the frequency band, the aperiodicity should be given for several frequency bands. Band aperiodicity distortion (BAP) is computed as aperiodicity over frequency bands. We measured the BAP distortion on band aperiodicity acoustic features using Equation 4.1. Aperiodicity had been widely used for speech coding applications, and speech analysis [Morise, 2016, Deshmukh and Espy-Wilson, 2003].

The periodicity in speech signal refers to periodically repeating speech units. As speech signals are almost periodic, speech signals inherently possess aperiodic speech units resulting in noise. From a speech production point of view, aperiodicity is characteristic noise added to the output of the vocal cord, resulting in responsible for distortion, raspiness, breathiness, jitter, and shimmer [Griffin and Lim, 1985,Griffin and Lim, 1988]. Therefore, aperiodicities are sounds that are unsynchronous with vocal fold motion.

Lf0 Root mean square error

The fundamental frequency is computed by a frame-by-frame estimation process using short-term analyses. A log of fundamental frequency provides useful insight over several orders of magnitude. The Root mean square error (RMSE) is computed between target and predicted log of $F0$ (Lf0), which is denoted as Lf0 RMSE, and defined as given below,

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=0}^{T-1}(Lf0_t - Lf0'_t)^2} \tag{4.2}$$

where $Lf0_t$ and $Lf0'_t$ are respectively the original and the predicted values of log of f0 values at the originally voiced frame t [Zangar et al., 2019].

4.2.2 Cosine Similarity

In this section, we present objective evaluation metrics to evaluate the performance of expressivity transfer, for which we computed speaker similarity score and expressive similarity score. First, we will discuss the cosine similarity and then its usage in defining speaker similarity and expressive similarity. The cosine similarity measures the similarity between vectors computed as the cosine of the angle between them. The cosine similarity is also equivalent to the inner product of the vectors normalized to both have a length of 1. The cosine of 90° is zero and is less than 1 for any angle in an interval of $[0° - 180°]$. The cosine angle is estimating the orientation of vectors and not the magnitude. Furthermore, the cosine similarity is often used to have a normalized score between -1 and 1.

The cosine similarity as a cosine angle is defined as given below in Equation 4.3. As the cosine of angle measurement gets closer to 1, then the angle, θ between the two vectors A and B is smaller, where A and B are vectors for which we want to estimate the cosine similarity. The cosine similarity is popularly used for measuring similarity in various domains such as computer vision [Nguyen and Bai, 2010], speaker verification [Bai et al., 2020], data mining [Usino et al., 2019, Eminağaoğlu and Gökşen, 2020]. For instance, the network is trained to optimize a generalized end-to-end speaker verification loss so that embeddings of utterances from the same speaker have high cosine similarity, while those of utterances from different speakers are far apart in the embedding space [Jia et al., 2018, Kim et al., 2021].

$$Cos(\theta) = \frac{A \cdot B}{\parallel A \parallel \parallel B \parallel} \tag{4.3}$$

We applied cosine similarity metrics to estimate speaker similarity scores and expressive similarity scores. We estimated speaker similarity using the cosine angle between embeddings of expressive synthesized speech and mean embeddings of the target speaker. Likewise, the expressive similarity is estimated as a cosine angle between expressive synthesized speech and mean embeddings of target expressivity.

As a first step toward computing similarity metrics, we developed two recognition systems based on a neural network for classifying speakers and emotions. We extracted embedding from the last layers of recognition systems during the evaluation phase, then computed similarity scores using the cosine angle. We implemented a convolutional recurrent neural network model for recognition tasks trained using the cross-entropy loss function. The pre-computed mean of each class of recognition systems is compared with embedding extracted for synthesized speech.

Recognition system

The recognition system is based on a convolutional recurrent neural network composed of convolutional and LSTM layers for extracting spatial and temporal information, as shown in Figure 4.1. We have used a sampling rate of 22050Hz and extracted Mel spectrograms as input acoustic features to an encoder network. We used 80 Mel filters and applied STFT with a window size of 1024 and a hop length of 256. The convolutional layer facilitates a preprocessing of input acoustic features, implemented with a

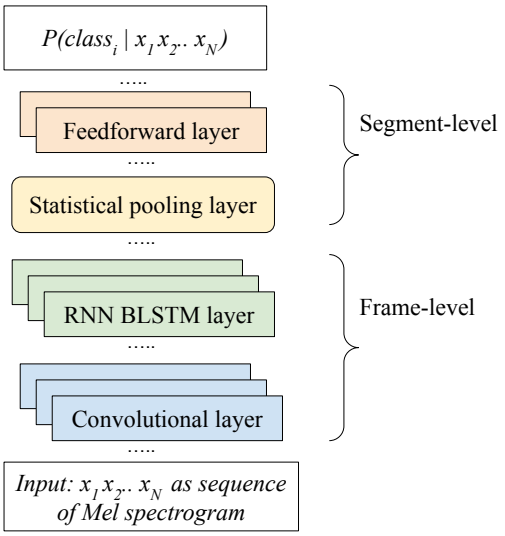

Figure 4.1 – The architecture of the speaker and expressivity recognition systems used to derive speaker and expressivity embeddings.

kernel size of 3 and stride equal to 1. These hyperparameters make the convolutional layer equivalent to the implementation of a time-delay neural network [Kreyssig et al., 2018]. Moreover, they are also used in creating x-vector embeddings [Snyder et al., 2018]. We used 4 LSTM layers with 256 hidden units. The hidden outputs of LSTM layers are given to the statistical layer. The statistical pooling layer calculates the mean vector μ as well as the second-order statistics as the standard deviation vector σ over frame-level features h_t $(t = 1, .., T)$ [Okabe et al., 2018]. Therefore, statistical pooling assists the recognition system in capturing long-term dependencies aggregated over variable-length speech segments. These aggregated statistics are concatenated and passed to two additional hidden layers with dimensions 512 and 256, with rectified linear units as non-linear activation functions. This neural network architecture is trained with cross-entropy loss to measure the total entropy between desired and predicted distributions of classes. We trained a speaker recognition system to extract the speaker embedding and estimate the speaker similarity score based on the cosine angle between reference speaker embedding and speaker embedding of synthesized speech. We used an emotion recognition system similar to the speaker recognition system to extract the emotion embeddings and measure the expressivity similarity score.

Expressive similarity score

We trained an emotion recognition system for obtaining an expressive similarity score with six emotions and neutral, totaling nine expressivity classes. We included six emotion classes from Ekman's theory of emotions [Ekman, 1999] and two more emotion classes, namely, amused and sleepiness [Adigwe et al., 2018]. We trained the emotion recognition system on the speech data from all the datasets described in Section 3.2 including for

French and English languages. The emotion recognition system is trained for 100 epochs with Adam optimizer [Kingma and Ba, 2015].

During the evaluation, expressive embedding is extracted for speech utterance as an output of the last layer corresponding to 256-dimensional embedding vectors. The extracted embedding x_e is compared with a mean of embeddings of desired emotion class, $x_{e,mean}$ using cosine angle, as stated in Equation 4.4,

$$Expressive\,similarity\,score = \frac{x_e \cdot x_{e,mean}}{\parallel x_e \parallel \parallel x_{e,mean} \parallel} \tag{4.4}$$

Speaker similarity score

The same architecture is also used for developing a speaker recognition system. We used all the speech synthesis datasets stated in Section 3.2 for the training of the speaker recognition system. In total, 145 speakers were included as output classes, which also includes variations in emotions for speakers of expressive speech synthesis datasets. Hence, the speaker recognition system could apprehend the speaker representation independent of the expressivity class.

The system is trained for 100 epochs with an Adam optimizer. During the evaluation, speaker embedding is extracted for speech utterance as an output of the last layer as 256-dimensional embedding. This extracted embedding x_s is compared with a mean of embeddings of desired speaker class, $x_{s,mean}$ using the cosine angle stated in Equation 4.5.

$$Speaker\,similarity\,score = \frac{x_s \cdot x_{s,mean}}{\parallel x_s \parallel \parallel x_{s,mean} \parallel} \tag{4.5}$$

4.3 Subjective evaluation

Irrespective of what TTS system is used, subjective evaluation is a critical stage in developing and deploying the TTS system. The subjective evaluation is also used in quality measures in telephone transmission, speech, and video quality transmission [Rix et al., 2001]. The quality of synthetic speech is a significant factor in the user experience in applications integrated with human-machine interactions. The common practice among researchers usually is to conduct the subjective evaluation by randomly selecting a set of samples from the test set and the corresponding speech signals generated by a baseline TTS system and by the proposed one.

Several subjective evaluation approaches have been proposed in the context of speech applications, such as Mean opinion score (MOS) [Polkosky and Lewis, 2003], MUltiple Stimuli with Hidden Reference, and Anchor (MUSHRA) [Polkosky and Lewis, 2003, Schoeffler et al., 2018], as well as preference tests like AB, ABX [Dall et al., 2014]. In MUSHRA, the listener is presented with the reference (labeled as such), a certain number of test samples, a hidden version of the reference, and one or more anchors. The anchor(s) purpose is to make the scale closer to an "absolute scale", ensuring that minor artifacts are not rated as bad quality. In AB tests for multidimensional scaling (MDS), the question is, are the two samples the same or different. This can be in terms of naturalness, quality, or intelligibility. As in the ABX test, two samples are given, with an additional reference

speech utterance. The listener has to judge if A or B is more like X. Again, this can be in terms of naturalness, speaker similarity, and emotional quality. This section will overview MOS and variants of MOS ABX tests to measure TTS systems and expressivity transfer performance, respectively.

4.3.1 Mean Opinion Score

The Mean opinion score (MOS) is popularly used in listening tests to evaluate either naturalness or intelligibility, or several qualitative factors in the synthesized speech [Wang et al., 2017b,Theis et al., 2016]. The Mean Opinion Scale (MOS) is a widely used method for evaluating the quality of telephone systems and synthetic speech, recommended by the International Telecommunications Union [Schmidt-Nielsen, 1992]. ITU 1994 recommends MOS with seven different questions. The MOS is a Likert-scale [Joshi et al., 2015] questionnaire, typically with seven 5-point scale items addressing the following speech characteristics, global impression, listening effort, comprehension problems, speech articulation, pronunciation, speaking rate, and voice pleasantness [Lewis and Hardzinski, 2015]. The main advantage of the MOS listening tests is the ability to provide quick feedback.

The survey is conducted in an uncontrolled environment. Hence information on the setup is collected to ensure the results can be put into perspective. Before starting the evaluation, each listener is asked if they are using in-ear headphones, over-the-ear headphones, desktop speakers, or laptop speakers. This procedure is adopted from the crowdMOS [Flavio et al., 2011] approach. As described in [Flavio et al., 2011], workers using loudspeakers generally have a smaller discrimination capacity than listeners wearing headphones. While participants are asked to use headphones during the test, this configuration is not enforced. With this metadata and the knowledge of the influence of the playback device, it is possible to analyze the results better.

In our thesis work, we designed MOS listening tests to evaluate synthesized speech's overall quality and naturalness. During the MOS listening tests, the listeners were provided with speech utterances and corresponding text. The listeners are supposed to provide a scale of 1 to 5, where one is poor and five is excellent. Furthermore, we provided instructions for the listener to conduct the listening tests in a noise-free environment. We also recorded the listener's name and French language level.

4.3.2 Expressivity Transfer

The main goal of this work is to transfer emotion as expressive attributes to the target speaker's voice without altering the speaker's voice characteristics. Due to a lack of reference expressive speech utterance, there is no possible way to extract quantitative results to evaluate the transfer of expressivity. This section presents speaker MOS and expressive MOS listening tests to measure qualitative evaluation of expressivity transfer.

In [Jia et al., 2018], the authors measured speaker similarity scores using a 5-point scale in the context of speaker adaptation in a multispeaker end-to-end TTS system. In our experiments, we have designed two independent listening tests to evaluate the speaker's voice in synthesized speech and how expressive synthesized speech is similar to target

expressivity. These listening tests are designed as combinations of MOS and preference tests such as ABX tests to evaluate the presence of characteristic features such as the speaker's voice or expressivity on a scale from 1 to 5. We used an absolute ranking scale to design speaker MOS as well as expressive MOS.

Speaker MOS

In speaker MOS listening tests, we paired each synthesized utterance with a randomly selected ground truth utterance from the same speaker and were instructed to provide scores from 1 (bad) to 5 (excellent). Each pair is rated by one rater with the following instructions: "You should not judge the content, grammar, or audio quality of the sentences; instead, just focus on the similarity of the speakers to one another."

Expressive MOS

In expressive MOS listening tests, we paired each synthesized utterance with a randomly selected ground truth utterance from the same emotion and were instructed to provide a score on a scale from 1 (bad) to 5 (excellent). Each pair is rated by one rater with the following instructions: "You should not judge the content, grammar, or audio quality of the sentences; instead, just focus on the similarity of the expressivity to one another".

4.4 Discussion

In this chapter, we described several evaluation metrics in the context of TTS and expressivity transfer. The need for multiple evaluation metrics provided various aspects of speech quality such as emotional strength, speaker's voice quality and overall quality of speech. We conducted MOS listening tests for estimating the overall speech quality of synthesized speech.

Many approaches proposed for prosody transfer or expressivity transfer suffer from source speaker leakage, where the voice quality of the source speaker is fused to the target speaker in synthesized speech [Karlapati et al., 2020]. Due to a lack of reference expressive speech, we could not use the classical TTS objective metrics for evaluating the performance of expressivity transfer. Therefore, we proposed cosine similarity metrics to measure two speech attributes, the speaker's voice, and expressivity. One of the challenges in expressivity transfer is to preserve the target speaker's voice. Moreover, these similarity metrics show the capability of TTS systems to disentangle the expressivity and speaker characteristics.

In line with similarity metrics, we designed the listening experiments to measure particular speech attributes in the context of expressivity transfer. We modified the ABX listening tests protocol to provide the scores on the absolute ranking scale for specific speech attributes such as the target speaker's voice and expressivity.

Chapter 5

Parametric Expressive Text-to-speech Synthesis

5.1 Introduction

The paradigm of expressive speech synthesis has shifted dramatically in the previous two decades. For direct control over expressivity, earlier expressive voice synthesis systems used formant and diphone-based techniques. Statistical parametric speech synthesis techniques, on the other hand, offer a way to manage expressivity in TTS systems with implicit control. In the 1990s, neural networks were already being used to learn the relationships between linguistic and acoustic features [Weijters and Thole, 1993, Cawley and Noakes, 1993, Tuerk and Robinson, 1993], as segment duration models [Riedi, 1995], and to extract linguistic features from raw text input [Karaali et al., 1998]. For neural network modeling, the main differences between now and the 1990s are more hidden layers, more training data, more advanced computational resources, more advanced training algorithms, and significant advancements in the various other techniques required for a complete parametric speech synthesizer: the vocoder and parameter compensation/enhancement/post-filtering techniques [Wu et al., 2016]. In the context of parametric TTS, DNN can be viewed as a replacement for the HMM-based decision trees used in the framework [Wu et al., 2016, Watts et al., 2016].

The main focus of this chapter is the utilization of the parametric TTS framework in two main tasks. First, to synthesize speech for a given input text in various emotions as expressivity. The other task is to transfer expressivity to Lisa, Siwis, and Tundra speaker's voices in the text to speech framework. In Section 5.2, we develop single speaker parametric TTS systems. These systems are explicitly trained for each speaker and each emotion, as available in the training data. Section 5.3 discusses proposed parametric TTS frameworks for expressivity transfer in a multispeaker setting. We propose to create the latent space representation for speaker and expressivity, which was used to transfer the expressivity by selecting a latent variable from desired speaker and expressivity class.

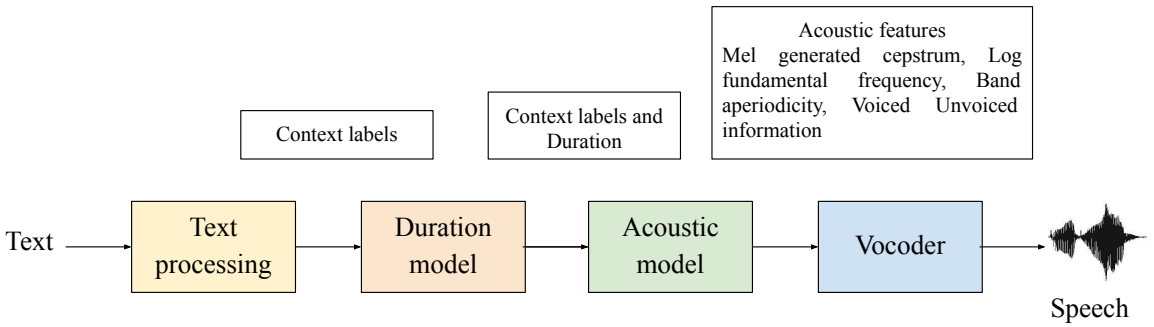

Figure 5.1 – Single speaker parametric TTS framework

5.2 Single Speaker TTS

The approach of synthesizing voice from text using the estimate of parameters representing speech signal is called a parametric Text-to-Speech system. Figure 5.1 illustrates the general framework of single speaker parametric TTS. The first module in Figure 5.1 is text processing, which creates a phonetic representation of text and provides linguistic and paralinguistic information. The speech parameter prediction module consists of duration and an acoustic model. The parameter prediction module employs a model to predict acoustic features as speech parameters like fundamental frequency, spectral parameters, duration, aperiodicity, etc. These speech parameters are provided as input to the vocoder to synthesize the speech waveform. HMM-based parameter prediction and DNN-based parameter prediction are the two most common types of parameter prediction frameworks [King, 2011, Fan et al., 2014, Zen et al., 2013].

Deep neural network approaches directly map textual features (also known as context labels) to speech parameters, which have proved to be highly effective in learning the intrinsic properties of acoustic features. This section describes the details of a single speaker parametric TTS system built for the French language. Moreover, we developed expressive TTS systems for each expressivity class in a single speaker setting.

5.2.1 Model Architecture

As explained in Chapter 2, HMM-based statistical parametric speech synthesis use decision trees to map linguistic context labels extracted from a front end to probability densities of acoustic features. These densities are then used to predict the speech parameters fed to the vocoder to generate a synthesized speech waveform.

In DNN based approach, the decision trees and probability densities (HMM) are replaced by deep neural networks to enhance the model's ability to learn complexity in mapping [Zen et al., 2013]. The number of context labels depends on the length of the sentence and is always less than the number of associated output acoustic frames. Therefore, we need an explicit duration model to predict the phoneme duration to estimate the exact number of acoustic frames required for the speech waveform generation. In the deep

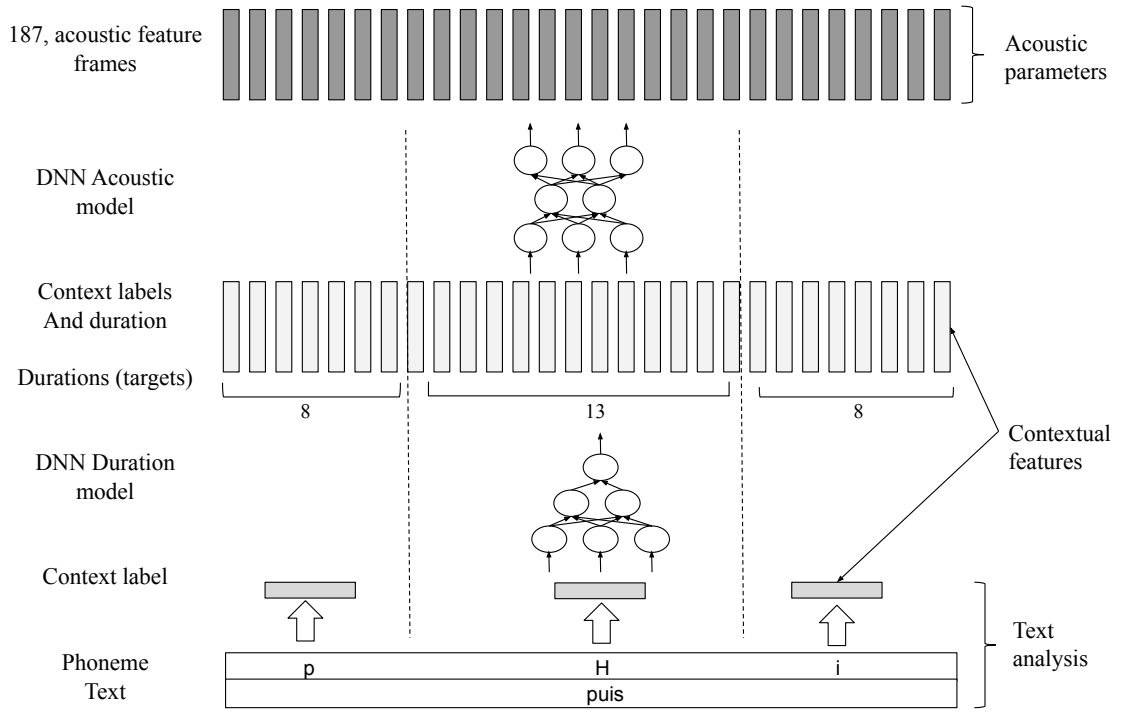

Figure 5.2 – Example of processing with the DNN based speech synthesis system with the duration model and the acoustic model.

neural network approach, speech synthesis modeling is divided into duration and acoustic models as demonstrated in Figure 5.2.

In the training phase, the text is converted into a sequence of context labels by the front end. Each context label contains information about phoneme durations to be predicted and context label features, as explained in Section 3.3. In the duration model, input contextual features for each phoneme are given to the neural network to predict duration output information. Thus, the duration model estimates the number of acoustic frames required for each phoneme as a regression problem. The duration model is trained independently with mean square error as a loss criterion to predict the phoneme duration. After that, contextual label information along with duration information is given to the acoustic model defined by a deep neural network architecture to generate output acoustic parameters, namely spectral parameters, aperiodicity, log fundamental frequency, and voiced-unvoiced flag.

Figure 5.2 represents the processing steps in the synthesis phase. After text analysis, for a given input contextual label features, durations are predicted with forward propagation through the neural network. Then the duration model's output and contextual features are given as input to the acoustic model to predict the acoustic parameters. Afterward, denormalization is performed on generated acoustic parameters using the mean and standard deviation values pre-computed from training data. Finally, the speech waveform is generated by applying a vocoder to the predicted acoustic parameters. The duration and acoustic models were implemented using bidirectional long short-term memory (BLSTM)

Table 5.1 – The objective evaluation results on single speaker single emotion parametric systems on French speech synthesis datasets

Speaker	Expressivity	MCD	BAP	Lf0 RMSE	Speaker similarity	Expressivity similarity
Lisa	neutral	**5.02**	1.36	20.09	0.84	0.97
Siwis	neutral	5.45	1.33	19.44	0.81	**0.98**
Tundra	neutral	5.92	**0.71**	**9.82**	**0.86**	0.96
Caroline	neutral	5.03	1.36	19.78	0.74	0.95
Caroline	anger	5.44	1.51	20.95	0.83	0.94
Caroline	joy	5.28	1.38	33.95	0.63	0.81
Caroline	sad	5.18	1.61	23.75	0.79	0.55
Caroline	fear	5.35	1.24	17.89	0.82	0.93
Caroline	disgust	5.08	1.11	22.51	0.77	0.47
Caroline	surprise	5.06	1.34	41.98	0.55	0.74

layers, where BLSTM is a type of recurrent neural network. We created a specialized speech synthesis system for each expressivity class for expressive speech synthesis.

5.2.2 Experimental Setup

We developed single speaker single emotion parametric TTS systems for the Caroline expressive dataset, Lisa neutral dataset, Siwis neutral dataset, and Tundra neutral dataset (neutral and six emotions). The parametric TTS systems were trained for 50 epochs with a mini-batch size of 64 and a 0.1 dropout rate. We implemented an early stopping mechanism for ten epochs that would end the training if the loss on the validation set did not improve during ten epochs. With 256 hidden units, we used five BLSTM layers. We utilized the Adam optimizer with a learning rate of 0.001. The speech signal was sampled at a rate of 16 kHz. Each speech dataset was split into 80, 10, and 10 train, validation, and test sets.

Section 3.3 and 3.4 explained the data preparation process to create text and speech features for the training parametric TTS system. Here, we use 180 contextual label features as input for the duration model, which generates the duration parameters for each sound unit (phoneme or pause). The same 180 input contextual labels, plus nine phone duration information and sub-phone features, were used to train the acoustic model. For each 5ms time frame, 187 acoustic parameters were calculated, including Mel generalized cepstrum (MGC) coefficients, log of fundamental frequency (Lf0), band-aperiodicity (bap), and voiced-unvoiced information (vuv).

5.2.3 Results

The performance of single speaker single emotion parametric TTS is measured with objective evaluation metrics as stated in Table 5.1. We conducted the error analysis for

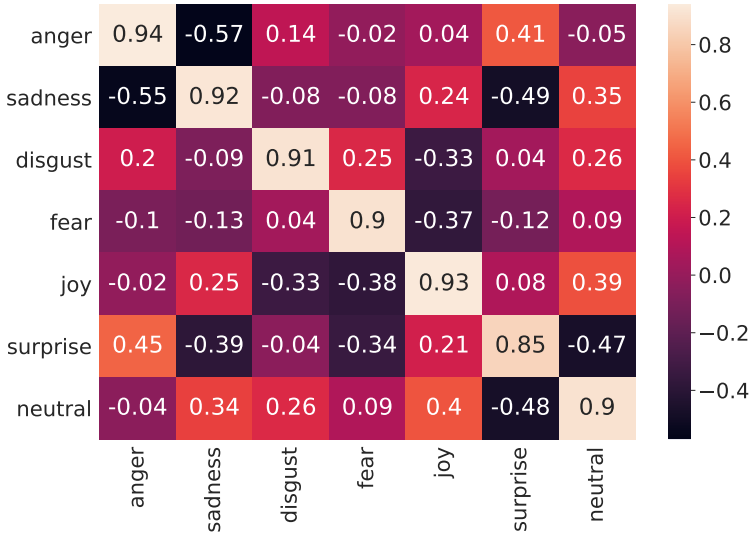

Figure 5.3 – Scores illustrating closeness for expressive similarity score computed on expressive speech synthesis results of Caroline expressive dataset

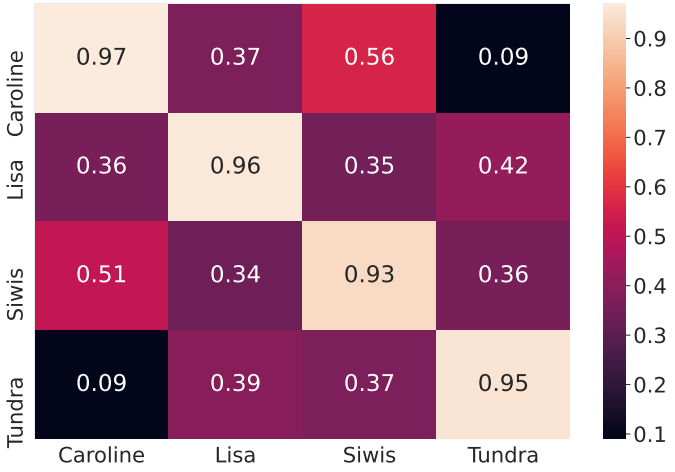

Figure 5.4 – Scores showing closeness for speaker similarity score computed on speech synthesis results of French speech datasets

the objective evaluation using acoustic features estimated by the acoustic model of the parametric TTS system and by the vocoder computed on the original speech data set. The objective measures are computed using the test set utterances. The results from Table 5.1 suggest that all parametric TTS systems, except Tundra, obtained BAP distortion above one. Hence, BAP distortion provides insight into the effect of the male and female speakers on the aperiodicity of synthesized speech.

The higher Lf0 RMSE scores for joy and surprise emotions reflect the inability of the acoustic model to learn the excitation of the Lf0 parameter, which is present in surprise and joy compared to other emotions. Furthermore, surprise and joy obtained the lowest speaker similarity scores, also inlined with the highest Lf0 RMSE scores obtained on these

two emotions.

Sad and disgust expressive parametric TTS systems obtained the lowest expressive similarity score, suggesting difficulty to model expressivity classes with lower energy and excitation levels. From Figure 5.4, anger and surprise are the expressivity classes that are closely related emotions due to higher energy and excitation acoustic features present in a speech signal. Russell's circumflex model explained in Section 1.2, both anger and surprise expressivity classes have higher activation. The results displayed in Figure 5.4 validate the correlation between anger and surprise expressivity classes. We plotted speaker similarity scores for synthesis results from each single speaker TTS system in Figure 5.3. The behavior of speaker similarity reflects that Caroline and Siwis have closely related speaker attributes.

5.3 Multispeaker TTS

In this section, we proposed architectures for implementing multispeaker expressive parametric TTS systems. Before that, we will explain the general framework of the multispeaker parametric TTS system. Multispeaker TTS system consists of a model trained using datasets from multiple speakers. An appropriate speaker code is selected to synthesize speech in the target speaker's voice [Hojo et al., 2016]. The simplistic approach to implement multispeaker TTS is by explicitly providing speaker code to the duration model and to the acoustic model. This speaker code helps the model learn the underlying output distribution associated with each speaker.

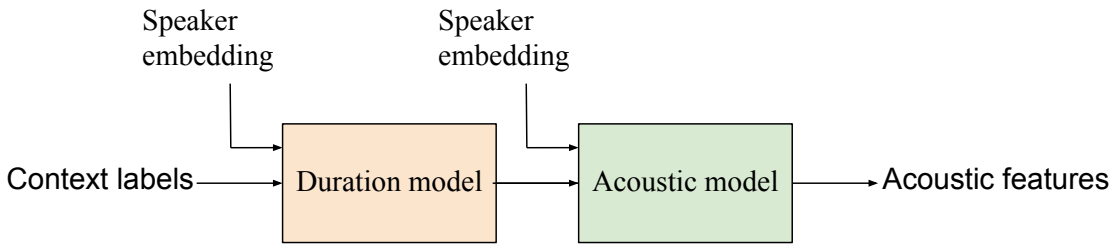

Figure 5.5 – Multispeaker parametric TTS framework

We proposed providing speaker embedding along with the context labels as input to the duration model. After that, context labels, speaker embedding, and phoneme duration are given to the acoustic model as input to predict the acoustic parameters of the given speaker, as depicted in Figure 5.5. The speaker code is either provided as a one-hot representation or by speaker embedding. The drawback of using speaker code is it enforces fixed dimensional model parameters. Therefore, if we need to add a new speaker voice into the TTS system, we will require to retrain the TTS system from the start. On the other hand, speaker embedding only needs to set the dimension of speaker embedding once; this enables to add the new speaker's voice without retraining the TTS system.

from scratch. In [Yang et al., 2016], authors proposed to use i-vectors as a replacement for the one-hot representation of speaker code.

In this section, first, we will overview the proposed approach for creating speaker embedding in the context of multispeaker TTS in Section 5.3.1. After that, we explain the proposed architectures for expressivity transfer in the parametric TTS framework. Section 5.3.3 describes the experimentation setup and demonstrates the proposed architecture's results. We conducted experimentation using two methods for performing variational inference. Additionally, we propose to improve the latent representation of expressivity using the metric learning technique.

5.3.1 Speaker Embedding

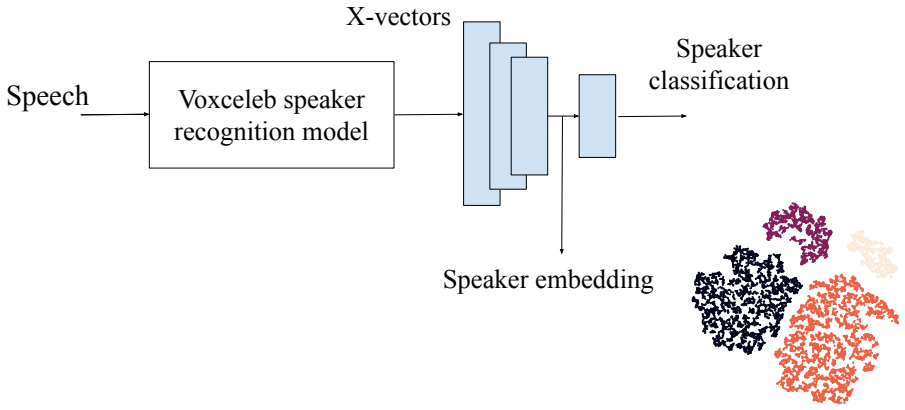

Figure 5.6 – X-vector based speaker embedding creation stating the protocol for extracting speaker embedding from pretrained speaker recognition model, and t-SNE plot of extracted embedding from speaker classification network on French speech synthesis datasets

We created speaker embeddings from the pretrained speaker recognition model to capture the speaker information. These embeddings should represent speaker characteristics irrespective of the textual content. We develop a speaker encoder network from a speaker recognition model trained on French speech synthesis datasets to generate such embeddings. Later, we use this speaker encoder to derive the speaker embedding.

The x-vector embeddings are deep neural network embeddings trained on time-delay neural networks with a statistical pooling layer trained for the speaker recognition task [Snyder et al., 2018]. Firstly, we extracted x-vector embeddings from the speaker recognition model trained on the Voxceleb dataset and available in the Kaldi toolkit [Chung et al., 2018, Povey et al., 2011]. To adapt the speaker embeddings to French speakers, we used extracted x-vector embeddings to train a feedforward neural network-based speaker recognition model to discriminate between speakers of our French speech synthesis datasets. Even though the speaker encoder is not trained to capture speaker identity directly, experimentation with speaker embeddings has shown the capability to represent speaker characteristics in synthesized speech. The low dimensional probability distribution of speaker embedding is plotted with t-sne plots as shown in Figure 5.6, which suggests the

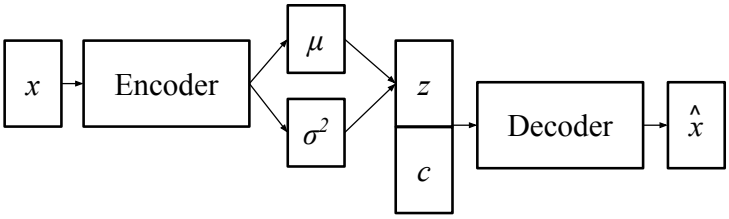

Figure 5.7 – RCVAE architecture used for training acoustic model. Here, x is a sequence of acoustic features to be reconstructed as $\hat{x}$, c is condition (textual features, speaker embedding) μ and σ^2 are mean and variance parameters provided by the encoder network, and used to generate the latent variable z.

distribution of embeddings for each speaker is distinctive, and each color represents one of the four French speakers, namely Siwis, Caroline, Lisa, and Tundra.

5.3.2 Proposed Frameworks

In this section, we propose three model architectures for the implementation of the acoustic model, namely variational autoencoder, variational metric learning, and Flow metric learning [Kulkarni et al., 2020a, Kulkarni et al., 2020b]. The common components in the above model architectures are the encoder network and decoder network. The encoder network is based on a variant of the recurrent neural network, thus enabling the processing of variable-length input acoustic features to create the fixed dimensional latent representation of expressivity. Furthermore, the decoder network is also implemented using a variant of the recurrent neural network to predict the variable-length acoustic features.

These proposed model architectures are developed in order to create the latent representation of expressivity. The expressivity transfer can be performed either by interpolating latent variables of desired expressivity or providing acoustic features of desired expressivity. In this work, we opt for expressivity transfer based on interpolation of latent variables of desired expressivity. Therefore, choosing an appropriate latent variable is a crucial factor in generating appropriate expressivity in synthesized speech.

We also implemented a duration model using DNN architectures based on variational autoencoder, variational metric learning, and Flow metric learning. However, these duration models were unable to perform the expressivity transfer for duration information in a multispeaker setting.

In this work, we used an explicit duration model to predict the number of acoustic feature frames required to synthesize the speech for a given text. Hence, we used a BLSTM neural network for predicting the duration for each phoneme, as explained in Section 5.2. More specifically, in the case of expressive speech synthesis, we used the duration model associated with desired expressivity to synthesize. The detailed experimentation with the duration model is explained in Section 5.2.

Model architecture: Variational autoencoder

In this section, we present the implementation of the acoustic model using recurrent conditional variational autoencoder (RCVAE) architecture. For the RCVAE architecture, we implemented a BLSTM based encoder network. The encoder's input is a sequence of acoustic features, x. The output of the last hidden state of the BLSTM layer is given to feedforward layers to estimate both the mean vector (μ) and variance vector (σ^2). The mean and variance are further used to describe the encoder's latent variable, z (using a reparameterization trick commonly used in variational inference). The detailed description of the variational autoencoder is explained in Section 2.3.1.

Similar to the encoder, the decoder network consists of BLSTM layers. The usage of BLSTM recurrency allows the model to extract long-term context from acoustic features. Also, it enables the processing of variable length acoustic features. The input of the decoder network is a latent variable z and the condition c. Here, the condition c corresponds to context labels as textual features, duration information, and speaker embedding. The decoder generates the sequence of predicted acoustic features $\hat{x}$, as shown in Figure 5.7. During the inference, we sample z from the latent space distribution. In the case of the expressive synthesis task, $z_{e,mean}$ of emotion, e is given as input along with condition c to synthesize the expressive speech of desired emotion, e.

The loss function in this model architecture corresponds to the reconstruction loss plus a regularization term defined with the Kullback-Leibler (KL) divergence. The reconstruction loss represents the expectation over the reconstruction of acoustic features, $logP(x|z,c)$, as $E_z[logP(x|z,c)]$ term. The KL divergence measure indicates how close the learned distribution $Q(z|x,c)$ is to the true prior distribution $P(z|c)$.

$$\mathcal{L}oss_{RCVAE} = E_z[logP(x|z,c)] + \lambda \cdot \mathcal{D}_{KL}[Q(z|x,c)||P(z|c)] \tag{5.1}$$

Recurrent network VAE frameworks often lead to a sudden drop in KL divergence [Bowman et al., 2016]. We added a variable weight, λ, to the KL divergence term as a KL annealing cost to deal with this problem. This assists in enhancing the disentangled latent space representation with good interpretability of the latent variable, as stated in the equation given below,

Model architecture: Variational Metric Learning

We propose improving the RCVAE architecture explained in the previous section by adding a second regularization term based on the metric learning technique. The variational metric learning (VML) term signifies the combination of variational inference-based KL divergence loss criterion and metric learning loss criterion. In VML architecture, we used multi-class N-pair loss to implement the metric learning loss criterion. A detailed explanation of metric learning is discussed in Section 2.4. We used the same neural network architecture as explained in the previous section to implement the RCVAE acoustic model.

The utilization of variational inference enforces the conditional distribution of the intraclass variance of expressivity classes, assuming that latent variables are sampled from a Gaussian distribution with zero mean and unit variance during the inference phase.

Therefore, the metric learning loss criterion reduces the intraclass variance of the distribution of latent variables for each expressivity class.

We use the mean of latent variables as a representation of emotion for expressivity transfer. Hence, the desired latent space should have well-separated clusters corresponding to the various expressivity classes. This indicates better clustering of emotion may lead to improved performance of expressivity transfer in the TTS system. Therefore, we propose to use multi-class N-pair loss in variational inference.

Multi-class N-pair loss has shown superior performance compared to triplet loss or contrastive loss by considering one positive sample and $N-1$ negative samples for N classes [Sohn, 2016]. Multi-class N-pair loss criteria increase the intercluster distance from $N-1$ negative samples and decrease the intracluster distance between positive samples and training examples. We employed the mean of latent variables of expressivity classes for mining the positive and the negative samples. In our case, positive samples refer to latent variables from the same expressivity class, and negative samples correspond to examples of different expressivity classes. For N expressivity classes, z^+ is a positive sample, and $\{z_i^-\}_1^{N-1}$ samples from negative classes as stated in Equation 5.2. This usage of multiple negative samples in training leads to faster model convergence, creating a robust representation of expressivity. The loss function for training VML architecture is as given below,

$$\mathcal{L}oss_{VML} = E_z[logP(x|z,c)] + \lambda \cdot \mathcal{D}_{KL}[Q(z|x,c)||P(z|c)] + \beta \cdot log(1 + \sum_{i=1}^{N-1} e^{z^\top z_i^- - z^\top z^+}) \quad (5.2)$$

We multiplied the multiclass N-pair loss with β factor to ensure the smoother convergence of model parameters. Additionally, we activated the multiclass N-pair loss after the first five epochs to have a warm-start for the training process of the acoustic model.

Model architecture: Flow metric learning (FML)

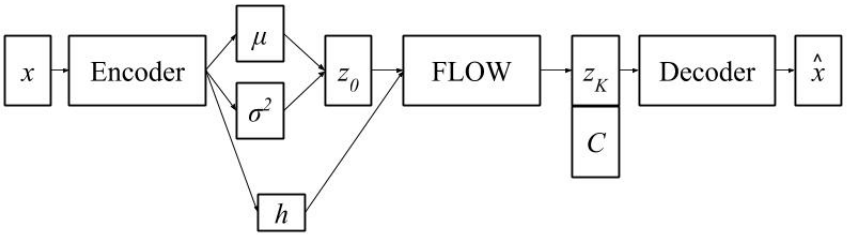

Figure 5.8 – Inverse autoregressive Flow architecture used for training acoustic model. x is a sequence of acoustic features to be reconstructed as $\hat{x}$, c is condition (textual features, speaker embedding) μ and σ^2 are mean and variance parameters provided by the encoder network, and used to generate the latent variable z_0. The z_K is obtained by passing z_0 through K IAF transformations

In the Flow metric learning (FML) acoustic model approach, we propose to improve the previous variational metric learning architecture. We perform variational inference based on normalizing Flow transformation instead of KL divergence term. The ability of variational inference performed using KL divergence is constrained due to intractable posterior distributions to be approximated by the class of known probability distributions, over which we search for the best approximation of the true posterior [Rezende and Mohamed, 2015]. The central issue in variational inference is the selection of approximate posterior distribution, which is usually assumed to be a Gaussian distribution with zero mean and unit variance. In the inverse autoregressive Flow, posterior distributions are formulated using a series of cascaded invertible transformations to map simple initial density to arbitrarily complex, flexible distribution with tractable Jocabians [Kingma et al., 2016]. We used IAF transforms to perform the variational inference for learning meaningful latent space representation of expressivity classes. The detailed explanation of IAF is discussed in Section 2.3.2.

The architecture for the IAF model has three components, namely encoder, IAF, and decoder, as shown in Figure 5.8. We implemented BLSTM layers for designing the encoder and decoder of the IAF acoustic model. The output of the encoder network is used to estimate the initial mean μ_0, initial variance σ_0^2, and hidden output h. Afterward, z_0 along with hidden output h is given to IAF transformation to obtain z_k after K transformations. We have conditioned the decoder network with Flow transformed z_k along with condition c corresponding to textual features, duration information, and speaker embedding. The decoder network generates predicted acoustic features $\hat{x}$ as an output. During the training phase, $\hat{x}$ is then used for computing the reconstruction loss, $logP(x|z,c)$. In the inference phase, we sample z_0 from latent space to obtain z_k after Flow transformation. Then z_k is given to the decoder network with condition c, to obtain acoustic features for speech synthesis using a vocoder.

The autoregressive structure of Flow allows simple computation of the Jacobian determinant of each transformation as a change in global posterior probability density of encoder network denoted as $logQ(z_K|x)$, where z_K is the output of the last Flow step. In this way, the flexible, tractable posterior distribution is created to perform the variational inference with the IAF.

For the implementation of Flow metric learning, the KL divergence loss term is replaced with a change in probability density, $logQ(z_K|x)$. Besides this, all the implementation details for the encoder-decoder network are kept as an RCVAE model. This includes the addition of multiclass N-pair loss and transferring expressivity using pre-computed latent variables, as explained in the previous section. The loss term used in training Flow metric learning architecture is given below,

$$\mathcal{L}oss_{FML} = E_z[logP(x|z,c)] + \lambda \cdot logQ(z_K|x) + \beta \cdot log(1 + \sum_{i=1}^{N-1} e^{z^\top z_i^- - z^\top z^+}) \qquad (5.3)$$

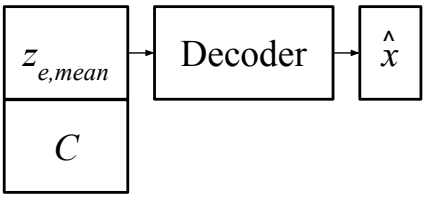

Figure 5.9 – Usage of the acoustic model during inference phase for multispeaker parametric TTS framework

Expressivity transfer with parametric TTS system

For the expressivity transfer task, we provided condition c as speaker embedding of the target speaker, context labels, and duration information obtained through the expressivity specific duration model. Afterward, we provided $z_{e,mean}$ of desired expressivity class e and condition c as input to the decoder network to obtain the acoustic features, $\hat{x}$. This process is illustrated in Figure 5.9.

5.3.3 Experimental Setup

For implementing IAF and RCVAE architectures, we built the encoder and decoder networks using 2 BLSTM layers of 256 hidden units each, a latent variable of 50 dimensions, a learning rate of 0.001, Adam optimizer initialized with default parameters, and a batch size of 10. For training the RCVAE model, we applied the λ of 0.001 to the KL divergence loss term, and a β multiplication factor of 1.0 is applied to the multiclass N-pair loss term. For the IAF model, we set λ to value 1, while the β factor applied to the multiclass N-pair loss term is set to 0.025 and incremented by 0.025 with each epoch. All the proposed architectures are trained for 50 epochs. The multiclass N-pair loss was activated after the first five epochs to ensure better convergence of the acoustic model parameters. We used the computed mean of latent variables for each expressivity class from the previous epoch in the training phase. These precomputed means are used in multiclass N-pair loss as positive and negative samples.

As mentioned before, we implemented a duration model explicitly for each speaker and expressivity class using a BLSTM network of 512 hidden units with the same configuration of batch size, learning rate, and optimizer as for the RCVAE architecture. We extracted a 512-dimensional x-vector using the speaker recognition model trained on the Voxceleb dataset [Chung et al., 2018] for speaker embeddings for all speech samples in datasets. Then, we implemented a five-layer of feedforward neural network and trained it to classify four French speakers (corresponding to our speech synthesis corpora) with (512-256-128-64-16) hidden units. This speaker recognition model was trained using cross-entropy loss criteria, Adam optimizer, and 50 epochs of training. We extracted speaker embedding for each speech sample by taking the output of activations of the last hidden layer of dimension 16.

We used four speech synthesis datasets for developing multispeaker expressive para-

metric TTS systems. The speech dataset used in this work are the Lisa speech dataset, Caroline expressive speech dataset, Siwis speech dataset, and Tundra speech dataset. Each speech dataset is divided into train, validation, and test sets in the ratio of 80%, 10%, 10%, respectively.

We parameterized speech using the WORLD vocoder [Morise et al., 2016] with 187 acoustic features computed every 5 milliseconds, namely 180 spectral features as Mel generalized cepstrum coefficients (MGC), 3 log fundamental frequencies (Lf0), 3 band-aperiodicities (bap) and 1 value for voiced-unvoiced information (vuv). We applied z-normalization on extracted acoustic features from the WORLD vocoder. For converting French text to linguistic features (also known as context labels, dimension 180), the Soja tool (internally developed in our team) is used as a front-end text processor.

5.3.4 Results

Table 5.2 – Evaluation metrics computed to measure the performance of parametric TTS system

Model	MOS	MCD	BAP	Lf0 RMSE	Speaker similarity	Expressivity similarity
RCVAE	2.62 ± 0.5	5.45	1.54	19.25	0.79	0.87
RCVAE+N-pair	2.97 ± 0.4	5.34	1.25	18.91	0.81	0.90
IAF+N-pair	**3.02 ± 0.4**	**5.21**	**1.24**	**17.84**	**0.81**	**0.91**

Table 5.3 – Evaluation metrics computed to measure the performance of expressivity transfer

Model	Speaker MOS	Expressivity MOS	Speaker similarity	Expressivity similarity
RCVAE	2.40 ± 0.2	1.53 ± 0.4	0.75	0.24
RCVAE+N-pair	2.86 ± 0.2	1.93 ± 0.3	0.76	0.26
IAF+N-pair	**2.93 ± 0.3**	**2.03 ± 0.3**	**0.77**	**0.27**

An informal listening experiment concludes that the IAF model without multiclass N-pair loss could not transfer the expressivity. Therefore, we excluded the IAF based acoustic model for evaluation. We used the objective evaluation metrics explained in Section 4.2 to measure the performance of proposed architectures. Furthermore, we also conducted a subjective evaluation using perceptive listening tests stated in Section 4.3. The speaker MOS, expressive MOS, speaker cosine similarity score, and expressive cosine similarity scores are used to evaluate the performance of the presented architectures for

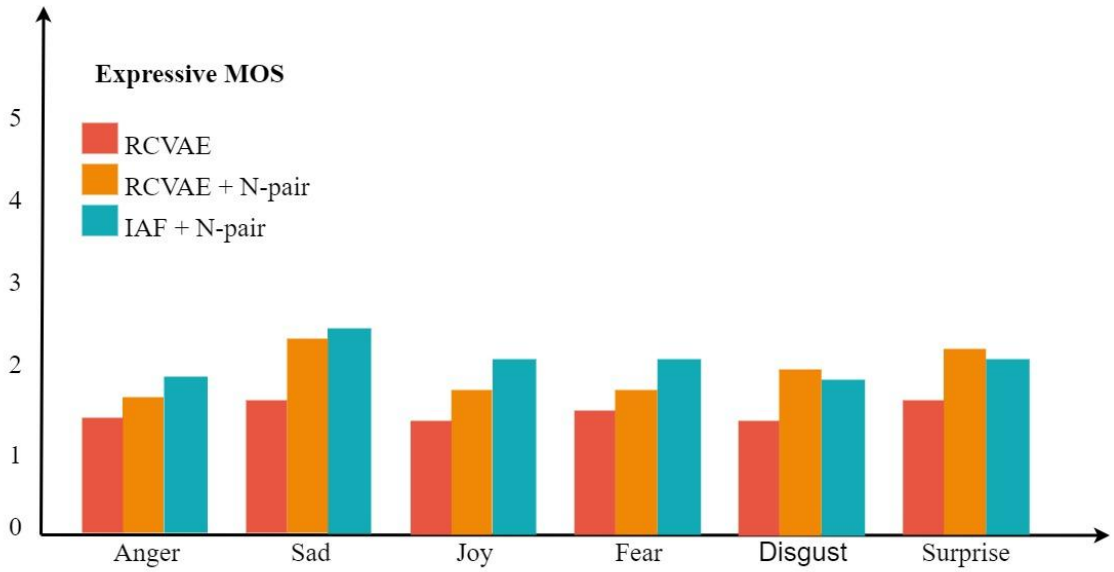

Figure 5.10 – Expressive MOS score bar plots on 6 emotions

the transfer of expressivity onto target speaker voices. The MOS, speaker MOS, expressive MOS are computed with associated 95% confidence intervals.

The MOS-based subjective evaluation was conducted with 12 French listeners. Each listener scored five stimuli for each speaker-emotion pair randomly chosen from the test set. Each listener scored three sets of stimuli for each target speaker-emotion pair to compute the speaker MOS and expressivity MOS for estimating the performance of expressivity. The linguistic contents of the speech stimuli and reference stimuli are not the same during the evaluation. We used the reference stimuli from the speech synthesis dataset.

The obtained results from Table 5.2 show that IAF with N-pair loss outperformed the RCVAE based proposed architectures for the TTS system. Furthermore, MCD, Lf0 RMSE, scores in line with MOS scores obtained using subjective evaluation. The cosine similarity measure for speaker and expressivity obtained on TTS systems are presented in Table 5.2 for the TTS systems.

The performance of parametric TTS systems on expressivity transfer tasks is shown in Table 5.3. For subjective evaluation of expressivity transfer, we utilized speaker MOS and expressivity MOS, and for objective evaluation, we used speaker similarity and expressivity similarity. The presented architectures retained the speaker's voice in expressive synthesized speech for Lisa, Siwis, and Tundra speaker with poorly perceived expressivity. Table 5.3 show that speaker similarity scores for expressivity transfer are close to the TTS system, contrary to the expressivity similarity score. In addition to this, we created a similarity matrix for expressivity similarity scores (Figure 5.13) and for speaker similarity (Figure 5.14) on results obtained from IAF N-pair based parametric TTS system. Similarity matrix on expressivity similarity score clearly indicates the affinity of certain expressivity classes such as surprise and anger. Also, it reflects the disassociation between each expressivity class.

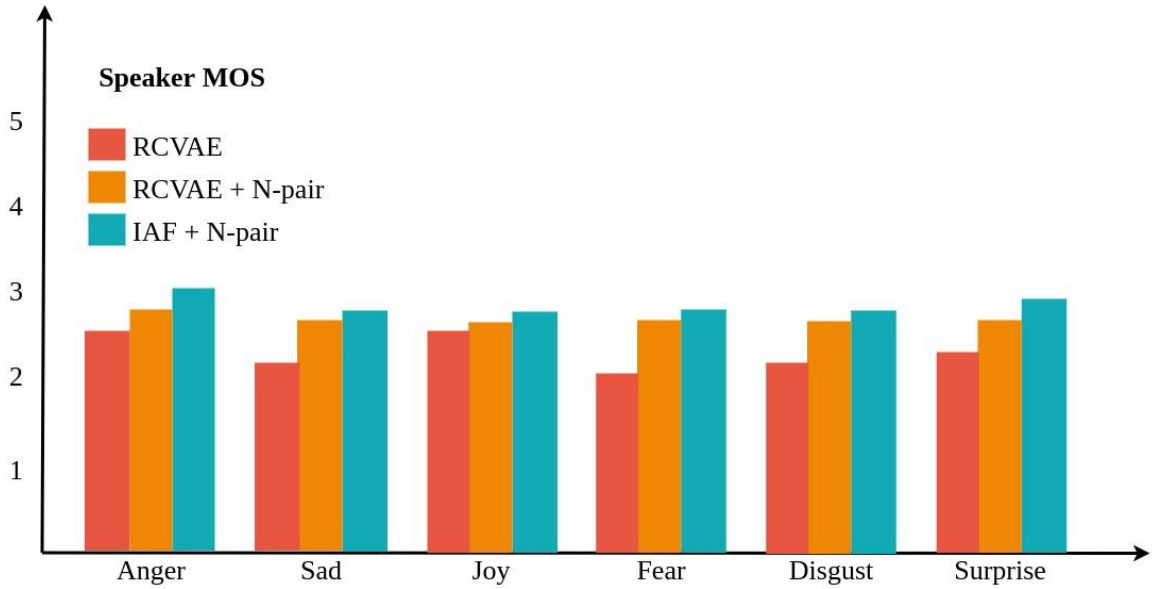

Figure 5.11 – Speaker MOS score bar plots on 6 emotions

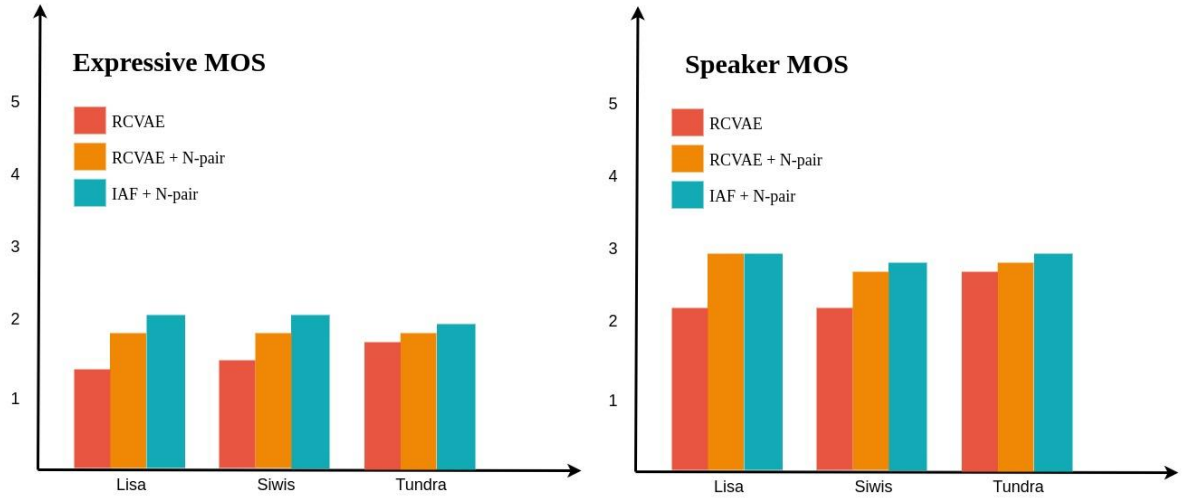

Figure 5.12 – Bar plots for Expressive MOS (left), and Speaker MOS (right) on speaker's for which only neutral speech data is available

From Figure 5.10 and Figure 5.11, the RCVAE model without N-pair loss performed poorly compared to other models. The speaker MOS and expressive MOS showed that while transferring expressivity, the addition of N-pair loss improves the retainment of the speaker's characteristics. Additionally, Tundra speaker shows that sadness and surprise are perceived as close to expressive characteristics for the original reference speech provided in evaluation. At the same time, anger expressivity class is the least perceived one for all speakers.

The bar plots presented in Figure 5.12 indicate the presented approach was equally able to transfer the expressivity from female (Caroline) to female (Lisa, Siwis) speakers as well as female (Caroline) to male (Tundra) speakers. The obtained results show that

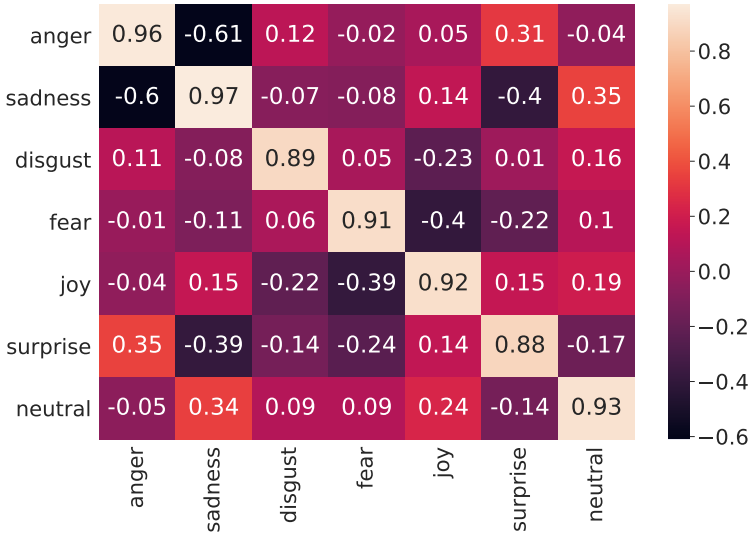

Figure 5.13 – Similarity matrix for expressive similarity score computed on IAF N-pair TTS system for speech synthesis datasets

Figure 5.14 – Similarity matrix for speaker similarity score computed on IAF N-pair TTS system for speech synthesis datasets

variational inference performed using IAF models improves the perceived expressivity in the desired speaker's voice due to the flexible, tractable posterior distribution. During the training phase, Lisa, Siwis, and Tundra had a different amount of training data was available. Irrespective of each speaker's different amounts of training data, From Figure 5.15, the t-SNE representation of the IAF N-pair model has tightly bounded clusters compared to the RCVAE N-pair model, which has more outliers for the cluster of expressivity classes. This t-SNE representation aligns with speaker MOS results and expressive MOS results. Moreover, the proposed objective evaluation metrics based on cosine similarity for the performance of expressivity are in line with the subjective evaluation measures.

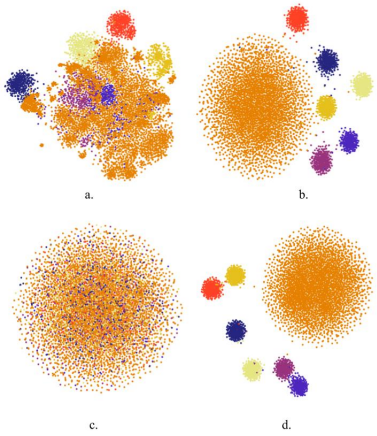

Figure 5.15 – t-SNE plot of latent representation of RCVAE acoustic model (a.), and RCVAE acoustic model with N-pair loss (b.); IAF acoustic model (c.), IAF acoustic model with N-pair loss (d.);Each color in t-SNE plot represents emotion; Here, neutral style by several speakers is represented by orange

5.4 Discussion

In a first approach, we developed a single speaker single emotion TTS system using the parametric TTS framework. As limited expressive speech data is available for Caroline speaker, it leads to poor performance on single speaker single expressive TTS systems compared to neutral TTS systems. For sadness and disgust expressivity classes, the cosine similarity-based expressivity scores obtained lower scores than other expressivity classes. This explains the difficulty in synthesizing the expressivity class with lower aperiodic excitation acoustic features. However, Russell's circumplex model states that the sadness and disgust expressivity classes are closely interlinked (refer to Section 1.2). Therefore, the selection of expressivity class and the relation between each expressivity class plays a vital role in developing expressive TTS systems.

In the presented multispeaker parametric TTS systems, speaker embeddings allow inheriting knowledge from the speaker recognition task in the TTS system. We fine-tuned the speaker encoder network on speakers from our French speech synthesis datasets. The speaker representation learned in such a way eases the convergence of the multispeaker TTS system. The perception tests show that the proposed approach retains the target speaker's voice while transferring the expressivity.

We developed parametric TTS systems that used deep metric learning and variational inference to learn the meaningful representation of expressivity, then used to transfer expressivity. We presented a variational autoencoder approach trained with multiclass N-pair loss to transfer expressivity in French's multispeaker text-to-speech synthesis. The multiclass N-pair loss function is used to disentangle expressivity information in the latent space. We use multiclass N-pair loss as the metric learning method because it considers information from multiple negative expressivity classes and a single positive expressivity class. The deep variational metric learning enforces the better clustering of expressivity in latent space representation.

For variational autoencoder's, the ability of variational inference is restricted due to intractable posterior distributions to be approximated by the class of known probability distributions, which is approximated to Gaussian distribution with zero mean and unit variance. The proposed IAF based parametric TTS system constructs posterior distributions using cascaded invertible transformation to tractable true posterior distributions. Hence, designing tractable variational families is an crucial problem in variational inference [Nalisnick et al., 2016, Salimans et al., 2015, Tran et al., 2015].

We improved the variational performance by adapting a series of inverse autoregressive transformations to map initial density to arbitrarily complex, flexible posterior distribution. We implemented an acoustic model using IAF trained and multiclass N-pair loss as a metric learning technique, named Flow metric learning. The presented work is the first approach that uses deep metric learning in an inverse autoregressive Flow based variational inference. The results show that the proposed IAF N-pair acoustic model enhanced latent space representation in a multispeaker expressive TTS system and outperformed the other approaches presented in the parametric TTS framework. The next chapter will overview end-to-end multispeaker TTS systems for expressivity transfer.

5.5 Conclusion

In this chapter, we presented three deep neural network models for the implementation of the acoustic model in a multispeaker parametric TTS framework. The aforementioned multispeaker parametric TTS framework leverages speaker embedding to allow a multispeaker expressive TTS system. We explicitly provide the speaker embedding in the multispeaker expressive TTS framework to create a latent space focusing on expressive information. We experiment with two methods: variational inference using Kullback–Leibler (KL) divergence and variational inference using inverse autoregressive Flow. We incorporated variational inference enabling neural networks to create meaningful expressivity representation in latent space. We interpolate the mean of latent variables of desired expressivity to transfer it to desired speaker's voice. This approach provides a way to synthesize expressive speech in the speaker's voice, for which only neutral speech is accessible. As clustering of latent variables plays a crucial role in the transfer of expressivity, we propose a metric learning methodology to differentiate the latent space in the parametric TTS framework. We discussed the results obtained using a multispeaker parametric TTS framework and the impact of proposed architectures on the transfer of expressivity.

Chapter 6

End-to-end Expressive Text-to-Speech Synthesis

6.1 Introduction

In [Glasmachers, 2017], end-to-end deep learning systems are defined as complex learning systems by applying gradient-based learning to the system as a whole. A single deep neural network model replaces all intermediate modules of deep learning-based systems pipelines. There have been significant advances in GPU-based processing resources in the last decade. As a result, increasingly complicated and stacked intermediate modules can be trained that focus entirely on predicting the desired output. Instead of a single objective function such as reconstruction loss or discriminative loss, multiple auxiliary objective functions are used to train end-to-end systems. The use of auxiliary objective functions that can affect sub-modules of the end-to-end model provides smoother convergence of model parameters and avoids unwanted local optima [Mirowski et al., 2017]. The approach based on end-to-end systems provides state-of-the-art results in various speech processing applications such as speech recognition [Amodei et al., 2016, Wang et al., 2020], speech-to-speech translation [Jia et al., 2021, Ren et al., 2020b], audio source separation [Luo and Mesgarani, 2019, Luo et al., 2019], etc.

In the context of end-to-end TTS system, intermediate modules of parametric TTS framework such as linguistic feature extraction as context labels, duration prediction model, and acoustic model are replaced with a single differentiable neural network. This neural network performs various subtasks such as encoding sequence of phonemes, generating alignment between text embedding and desired Mel spectrogram and post-filtering module for improving predicted Mel spectrogram, etc. In end-to-end TTS systems, text embedding refers to hidden representation created either from a sequence of characters or from a sequence of phonemes for given text input. Moreover, these approaches shown state-of-art results on standard speech synthesis datasets with significant improvement in MOS scores [Shen et al., 2018, Ping et al., 2017, Ren et al., 2020a].

This chapter presents the proposed end-to-end TTS system developed for expressivity transfer. Section 6.2 discusses proposed architectures based on the autoregressive Tacotron 2 system to transfer expressivity. Autoregressive TTS systems include coarse-

grained and fine-grained approaches. In Section 6.3, we propose non-autoregressive expressive end-to-end systems based on two deep generative models, namely, Generative Flow and Diffusion probabilistic models. Besides this, it also describes a multiscale expressivity encoder to create fixed dimensional expressivity embedding using multiple classical emotional acoustic features. The principal focus of this chapter is to investigate various end-to-end TTS approaches for expressivity transfer.

6.2 Autoregressive TTS

In this section, we present end-to-end TTS systems based on Tacotron 2 system [Shen et al., 2018]. The Tacotron 2 consists of a recurrent sequence-to-sequence acoustic feature prediction network that maps text embeddings to Mel spectrograms. The encoder-decoder neural network structure in these models converts a phoneme sequence to a sequence of acoustic feature frames [Sotelo et al., 2017, Arik et al., 2017, Wang et al., 2017b, Shen et al., 2018]. We extend the state-of-the-art Tacotron 2 model based on sequence to sequence with an attention module to implement an end-to-end multispeaker expressive TTS system. We extend the Tacotron 2 system by adding an expressivity encoder and a speaker encoder to function effectively with multiple speakers and expressivity.

In the autoregressive TTS framework, the decoder network receives input as sequential text embedding and attention weights as alignment. Mel spectrogram is then predicted frame by frame, taking into account previously predicted frames of Mel spectrograms. This strategy was developed primarily in the context of sequence-to-sequence learning.

In Section 6.2.1, we describe coarse-grained expressivity transfer systems, where a single time-independent expressivity embedding represents a latent variable. This Section gives details of the propose architecture for coarse-grained expressivity transfer leveraging on expressivity encoders based on global style token (GST) [Wang et al., 2018] and variational autoencoders (VAE) [Zhang et al., 2019]. In coarse-grained techniques, we also explored the addition of deep metric learning loss criterion to improve expressivity representation.

The fine-grained expressivity transfer framework, extracts time-dependent embeddings to produce an embedding representation [Karlapati et al., 2020]. Latent representations in fine-grained expressivity transfer are time-dependent at the phoneme and Mel-spectrogram frame level [Yang et al., 2020a, Sun et al., 2020b, Sun et al., 2020a]. We present multi-stage attention mechanism to derive phoneme-dependent expressivity representation, explained in Section 6.2.2.

6.2.1 Coarse-grained Expressivity Transfer

The primary goal of the coarse-grained technique is to provide a time-independent fixed dimensional embedding to represent expressivity. Expressivity interpolation is done in the same way that expressivity transfer is performed in Section 5.3.2. We extend the state-of-the-art Tacotron 2 system [Shen et al., 2018] based on sequence to sequence with attention module to implement an end-to-end multispeaker expressive TTS system. To

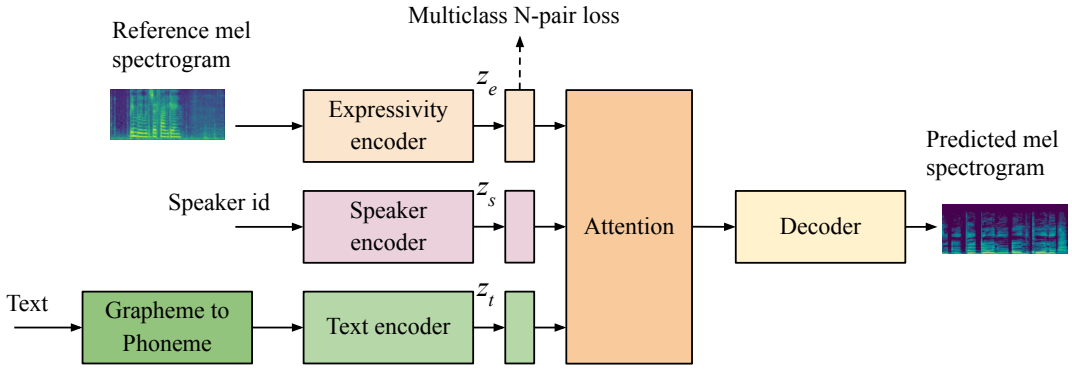

Figure 6.1 – End-to-end multispeaker expressive TTS system based on coarse-grained expressivity transfer

work with multiple speakers and expressivity, we modify the Tacotron 2 approach by adding an expressivity encoder and a speaker encoder.

The proposed end-to-end TTS system takes input as a sequence of phonemes, obtained through the grapheme-to-phoneme converter (explained in Section 3.3) for provided text. The first neural network module in end-to-end TTS systems is a text encoder, comprised of convolutional layers and Bidirectional Long Short Term Memory (BLSTM) layer, which creates z_t as an embedding representation of phoneme sequence. The reference Mel spectrogram is given to the expressivity encoder to extract the latent expressivity information as expressive embedding z_e. We will provide the implementation details of the expressivity encoder later in this section. For enabling the multispeaker setting, we provide the speaker identity to the speaker encoder to create embedding, z_s. The speaker encoder network maps the speaker index to non-linear fixed dimensional speaker embedding.

Following that, as illustrated in Figure 6.1, z_t, z_e, and z_s are concatenated and given as input to the location-sensitive attention module. This assists the end-to-end TTS in determining the alignment between the phoneme sequence and the desired Mel spectrogram. The decoder takes encoder outputs with attention vector as input to predict the Mel spectrogram frame by frame. The output from the previous frame is first passed through the pre-network (pre-net). The pre-net is composed of a fully connected recurrent network based on BLSTM layers which use the ReLU activation function. This predicted Mel spectrogram from the pre-net and recurrent network is further passed through the post-network (post-net). The post-net network is made up of 5 layers of a convolutional network. The post-net enhances the overall reconstruction performance of the Mel spectrogram by estimating residual to add to the predicted Mel spectrogram.

Expressivity encoder

Two neural network designs for expressivity encoder implementation were investigated: the Global Style Token (GST) [Wang et al., 2018] and the Variational Autoencoder (VAE) [Zhang et al., 2019]. The GST expressivity encoder consists of a reference encoder, style attention, and style embedding. The reference encoder converts a variable-length Mel

spectrogram into a fixed-length vector, subsequently given to the style attention layer as reference embedding. This layer uses a multi-head attention module as an expressivity encoder output to extract the similarity between reference embedding and each token in style embedding. In our work, style embedding z_e represents the expressivity as a stylistic factor to learn from the reference embedding.

The second design for the expressivity encoder is VAE-based, consisting of a reference encoder and two feed-forward layers to create the mean and standard deviation of the latent variable z_e. The z_e is computed using a reparameterization technique with mean and standard deviation. The Kullback Leibler (KL) annealing problem affects VAE-based frameworks, in which the KL loss term [Bowman et al., 2016] suppresses reconstruction loss. After a few early epochs, the KL divergence term in the KL annealing problem approaches zero. To avoid this, the KL loss is multiplied by an extra weight (close to zero), which is incremented throughout the training phase.

The end-to-end TTS system is trained with the multiclass N-pair loss criterion along with reconstruction loss and KL loss for variational inference. For the transfer of expressivity, we use pre-computed means of latent variables of each expressivity class. Thus, during the inference phase, the mean of expressivity embeddings is used to transfer expressive attributes to the target speaker's voice. The latent space representation of unclustered emotions may lead to the poor transfer of expressivity. Hence, for better performance of expressivity transfer, we need the tightly bounded representation of the expressivity embedding. Therefore, we propose a novel deep metric learning framework implemented using multiclass N-pair loss to further enhance the expressivity representation.

6.2.2 Fine-grained Expressivity Transfer

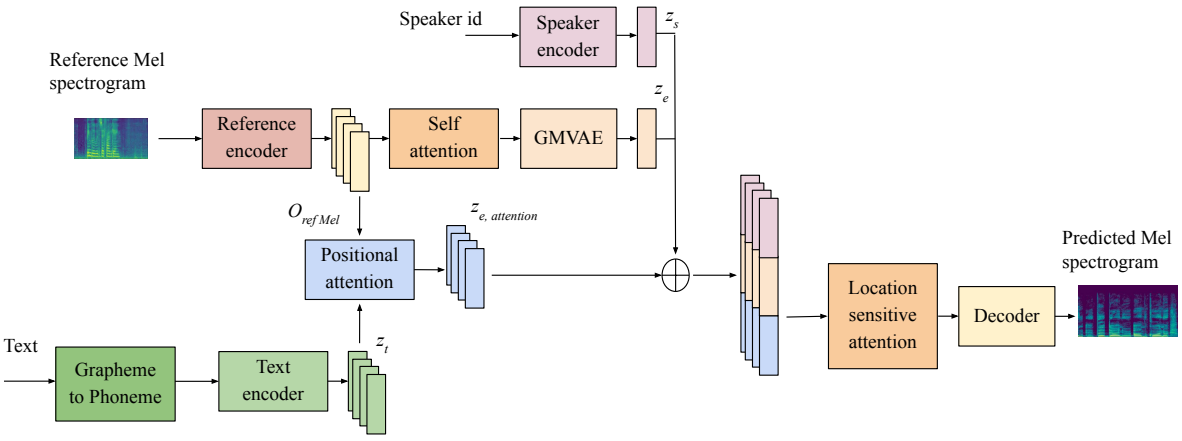

Figure 6.2 – Framework for multi-stage attention based fine-grained expressivity transfer in multispeaker TTS

The fine-grained expressivity based end-to-end TTS system allows learning phoneme or utterance dependent expressivity representation. As for the previous approach (Section 6.2.1), the proposed multispeaker Tacotron 2 decoder receives inputs in the form of

latent representations extracted from text, speaker identity, and a reference expressive Mel spectrogram. In this section, we propose to generate expressive attention weights using variable-length latent representations derived from reference Mel spectrograms and text embedding.

A reference encoder creates variable-length segmental latent representation from the input reference Mel spectrogram. We apply three attention strategies in the proposed work at differing stages of multispeaker expressive TTS. Then follows the self-attention layer [Vaswani et al., 2017], which incorporates salient expressivity information from the reference encoder's output. The self-attention layer's output is processed through the Gaussian mixture variational autoencoder (GMVAE) layer [Dilokthanakul et al., 2016] to extract a global representation of the expressivity class. The location-sensitive attention is utilized to generate expressive attention weights from the text encoder and reference encoder outputs. Depending on the phoneme sequence, this expressive attention weight provides the appropriate expressivity strength to be synthesized. Therefore, usage of location-sensitive attention enables the TTS system to incorporate time-varying expressivity information at a local level. Finally, the decoder employs location-sensitive attention to determine the alignment of text-Mel spectrogram pairings. These attentions at various levels of speech synthesis assist in the discovery of local and global expressivity characteristics.

For developing fine-grained expressive end-to-end TTS, we modified the model architecture described in Section 6.2.1. The proposed approach takes input as text, reference Mel spectrogram, and speaker identity. The text is mapped into a sequence of phoneme identity using explicit grapheme to phoneme converter. We used the same text encoder as used in Section 6.2.1 to create text embeddings, z_t. The fixed dimensional speaker embedding, z_s, is extracted from the speaker encoder for provided input speaker identities.

We extracted the expressivity information using a reference Mel spectrogram. The reference Mel spectrogram is passed to the reference encoder, which generates a segmental representation of expressivity, denoted as O_{refMel}. The reference encoder comprises six layers of stacked 2D convolutional layers with batch normalization. The gated recurrent units are used for recurrent pooling to compress variable-length O_{refMel} to a fixed dimensional. Then, it is passed through the self-attention layer to highlight salient expressivity features. We employed a hierarchical generative model based on multivariate Gaussian mixture variational autoencoder [Hsu et al., 2019] to disentangle the global representation of expressivity denoted as z_e. The GMVAE layer models expressivity as latent attributes using a mixture of Gaussian distribution. This allows the discovery of hidden expressive attributes and makes it easier to disentangle latent space.

For fine-grained expressivity transfer, we extract local information by obtaining attention weights as a correlation between text embedding and segmental representation, O_{refMel}. We integrated both representations using location-sensitive attention, which aligns the z_t and O_{refMel}. This attention output provides insight into expressivity strength for each sequence of text embedding. The output of attention is denoted as $z_{e,attention}$, which has the same length, L_{text} as text embedding.

The first proposed architecture fine-grained model I is shown in Figure 6.2. In case of fine-grained model I, speaker embedding, global expressive embedding, and expressive attention outputs are concatenated together to obtain encoder output vector from all

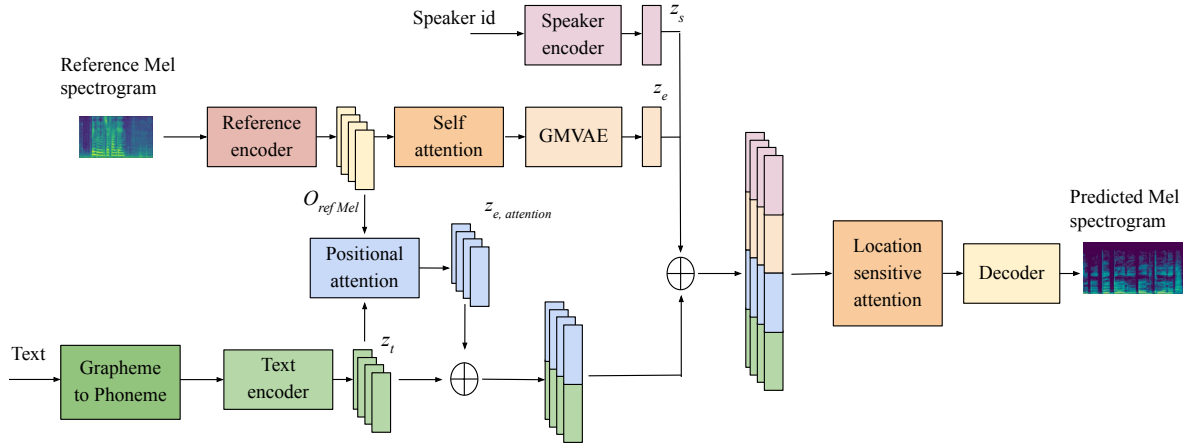

Figure 6.3 – Framework for multi-stage attention based fine-grained expressivity transfer in multispeaker TTS

encoders as $L_{text} \times (z_s, z_e, z_{e,attention})$. After that, we experimented with another architecture termed fine-grained model II. The main extension in fine-grained model II is that speaker embedding, global expressive embedding, expressive attention outputs, and text embedding are concatenated together to obtain encoder output vector from all encoders as $L_{text} \times (z_s, z_e, z_{e,attention}, z_t)$, as described in Figure 6.3.

We applied a third attention mechanism, location-sensitive attention, to align the target Mel spectrogram and encoder output. The decoder uses the encoder output to generate the Mel spectrogram frame by frame. The output from the previous frame is first passed through the pre-net. The pre-net is composed of fully connected layers with the ReLU activation function. The predicted Mel spectrogram from the pre-net and recurrent network is passed through the post-net. We used the same implementation of post-net similar to the proposed coarse-grained autoregressive TTS systems.

6.2.3 Expressivity Transfer

After training coarse-grained end-to-end TTS, we computed the mean of latent variables for each expressivity class. During the speech synthesis phase, we provided a speaker identity of the target speaker as input to the speaker encoder, a sequence of phonemes as an input to the text encoder, and a mean of latent variables of the target expressivity class. Therefore, we could predict the Mel spectrogram without the explicit need for reference Mel spectrogram as input to the expressivity encoder.

The fine-grained models depend on the reference Mel spectrogram for deriving expressivity representation at the global and local levels. Therefore, the choice of reference Mel spectrogram plays a vital role in the performance of fine-grained models. The expressivity transfer is categorized into parallel transfer and non-parallel transfer. In the case of parallel transfer, the reference speech utterance given to the expressivity encoder has the same textual content as the input phoneme sequence given to the text encoder (but a different speaker). On the other hand, the non-parallel transfer involves a mismatch of

textual content between the input given to the expressivity encoder and the input given to the text encoder (and also a different speaker). For non-parallel transfer, we selected reference acoustic features extracted from long speech utterances of desired expressivity. It is worth noting that the training was done for both cases in a parallel way.

6.2.4 Experimental Setup

We used 4 French female speech synthesis datasets for implementing our end-to-end multispeaker expressive TTS system. The speech datasets used are Lisa neutral speech dataset (approx. 3hrs, in house speech synthesis dataset), Siwis, neutral speech dataset (approx. 5hrs), Synpaflex speech dataset (approx. 7hrs), and Caroline expressive speech dataset. Caroline expressive speech dataset consists of 6 emotions, namely joy, surprise, fear, anger, sadness, and disgust (approx. 1hr for each emotion). Besides expressive speech, the Caroline speech dataset also has neutral speech recorded for approximately 3hrs. Each speech dataset is split into train, validation, and test sets in $80 : 10 : 10$ ratio, respectively. In the Synpaflex dataset, expressive speech samples are available. Still, due to an insufficient number of speech samples for each emotion and the unbalanced distribution of emotional speech samples, we have used only the neutral voice of Synpaflex in our work.

For coarse-grained autoregressive TTS systems, we used a sampling rate of 16000 and a WaveGlow neural vocoder trained on the available French speech synthesis datasets. In the case of fine-grained autoregressive TTS systems, we used a sampling rate of 22050 Hz and extracted Mel spectrograms as acoustic features to be predicted by an end-to-end TTS system. The speech waveforms are synthesized using WaveGlow neural vocoder for an input Mel spectrogram. We used the WaveGlow model pre-trained on the LJS dataset, as released on the official implementation website [12].

As input features to the end-to-end TTS, we have used a sequence of phonemes converted from the text. For French grapheme to phoneme conversion Soja tool (developed internally in the team) is used as a front-end text processor. A detailed explanation of preprocessing of text and postprocessing for extracting the Mel spectrogram is described in Chapter 3.

The text encoder is composed of three stacked convolutional layers of 512 filters each. The usage of the text encoder's convolutional layers with a kernel size of 5x1 facilitates the effect of N-gram on the input sequence of phonemes. Hence, the text encoder can understand contextual dependencies between sequences of phonemes. Batch normalization and ReLU activations are applied to the output of each convolutional layer. As explained earlier, the text encoder consists of convolutional layers and BLSTM layers. Therefore, the output of stacked convolutional layers is given to a BLSTM layer of 256 hidden units to create a time-dependent hidden representation as text embedding.

The location-sensitive attention is used to generate the alignment between the sequence of phonemes and the output sequence of the Mel spectrogram using 128-dimensional attention weights. The pre-net comprises two 256 hidden units feedforward layers, the output of which is concatenated with the attention context vector and passed through

12. https://github.com/NVIDIA/waveglow

two LSTM layers of 1024 hidden unit. Post-net consists of 5 stacked convolutional layers with 512 kernel sizes, batch normalization, and Tanh activation except for the last layer.

The reference encoder has been utilized as the first module to implement an expressivity encoder in all proposed autoregressive end-to-end TTS systems. The reference encoder consists of a stacked convolutional layer preceded by Gated Recurrent Units (GRU). We used a 2-D convolutional layer with a 3x3 kernel, stride of 2, batch normalization, and ReLU activation function. We used filters of sizes $32, 32, 64, 128$ and 128 for six convolutional layers, respectively. The output of convolutional layers is passed through GRU layers with 128 hidden units. The output of the last hidden state is extracted as reference embedding.

The GST expressivity encoder is implemented by passing the output of the reference encoder, reference embedding through multi-head attention with eight heads of attention. The reparametrization method is employed in the VAE expressivity encoder by passing reference embedding to two feedforward layers to represent mean and standard deviation. We have used a 128-dimensional latent variable of expressivity for both GST and VAE. In the case of VAE based expressivity encoder, we multiplied the KL loss term with a weight of 0.00001 for every 200 training steps till 150k iterations to avoid the KL annealing effect. After that, we increased the weight by 0.0001 after every 200 iterations until the weight factor reached the value of one. We incorporated a similar technique for fine-tuning with multiclass N-pair loss, for which until 150K training steps, a weight of 0 is applied to the multiclass N-pair loss, and afterward, the weight is increased by 0.001 after every 200 steps. Thus, multiclass N-pair loss is activated only after reconstruction loss starts converging towards the global minima direction. Hence, avoiding overfitting on discriminative loss on expressivity encoder. All the presented autoregressive TTS systems are trained for 500k iterations.

We used the same Tacotron 2 configuration explained above for implementing end-to-end TTS systems based on fine-grained expressivity transfer. The self-attention module comprises eight heads for multi-head attention, which is then given to the GMVAE layer to create global expressivity embedding of 256 dimensions. The same strategy as the VAE expressivity encoder is used for training fine-grained architectures for weight applied to KL divergence. For fine-tuning the fine-grained expressivity TTS systems, we multiplied the KL loss term with a weight of 0.0001 every 200 steps for 150k iteration steps. After which, the weight factor is incremented by 0.001 after every 200 iterations.

6.2.5 Results

We used the objective evaluation metrics explained in Section 4.2 to measure the performance of proposed architectures. Furthermore, we also conducted a subjective evaluation using perceptive listening tests described in Section 4.3. The speaker MOS, expressive MOS, speaker cosine similarity score, and expressive cosine similarity scores are used to evaluate the performance of the presented architectures for the transfer of expressivity onto target speaker voices. The MOS, speaker MOS, and expressive MOS are computed with associated 95% confidence intervals.

Table 6.1 – TTS Evaluation metrics computed to measure the performance of autoregressive end-to-end TTS systems

Model	MOS	MCD	BAP	Lf0 RMSE	Speaker similarity	Expressivity similarity
E2E GST	3.51 ± 0.3	4.36	0.83	18.23	0.91	0.90
E2E VAE	3.38 ± 0.4	4.76	0.88	18.79	0.92	0.91
E2E GST N-pair	3.72 ± 0.4	4.33	0.73	18.18	0.92	0.90
E2E VAE N-pair	3.47 ± 0.3	**4.21**	**0.72**	18.57	**0.93**	0.93
FG model I	3.81 ± 0.2	4.49	0.82	16.68	0.81	0.93
FG model II	**3.85** ± 0.2	4.50	0.83	**16.66**	0.83	**0.94**

TTS system

In this section, we will discuss the evaluation of proposed TTS systems for text-to-speech synthesis task. The MOS-based subjective evaluation was conducted with 12 French listeners. Each listener for MOS tests scored 18 stimuli for each autoregressive TTS system. For each session, we randomly shuffled the speech stimuli in the listening test.

Table 6.1 details the performance of autoregressive TTS systems on objective evaluation metrics, MCD, BAP, Lf0 RMSE, and subjective evaluation metric, MOS. From Table 6.1, the addition of multiclass N-pair loss improved the performance of coarse-grained end-to-end TTS system on both expressivity encoders, GST and VAE. E2E GST N-pair system received high MOS scores among coarse-grained TTS systems. Thus usage of multi-head attention assists in the speech synthesis task. The obtained MOS scores for each TTS system in Table 6.1 are consistent with the objective evaluation results. MCD and BAP metrics for fine-grained TTS systems are slightly higher than coarse-grained TTS systems. But, both fine-grained TTS systems received the lowest Lf0 RMSE scores. Consequently, fine-grained autoregressive TTS systems performed best in subjective evaluation based on MOS listening tests, with FG model II outperforming all other autoregressive systems.

Furthermore, we created a similarity matrix for expressivity similarity scores (in Figure 6.4) and speaker similarity scores (in Figure 6.5) on speech synthesis results of FG model II. This matrix represents the similarity scores obtained with respect to other classes of expressivity or speaker. For example, for the speaker similarity matrix, we obtained scores illustrating similarities between various speaker's synthesized speech.

The matrix representing similarity scores for various expressivity classes (Figure 6.4) demonstrates the closeness of expressivity class anger, joy and surprise, while other expressivity classes are distant. Thus, it allows us to observe the capability of the FG model II TTS system to synthesize expressive speech with well-segregated expressivity classes, i.e., lesser disambiguation between expressivity classes. The matrix for speaker similarity scores provides information regarding FG model II's capability to learn the speaker

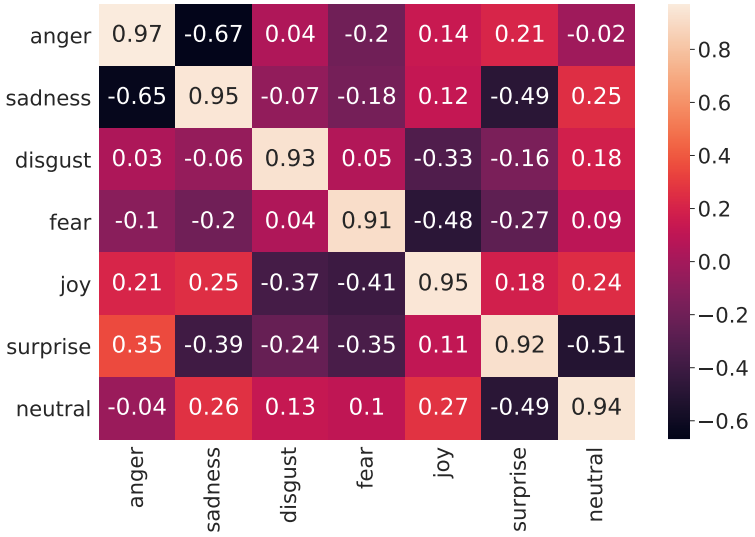

Figure 6.4 – Similarity matrix for expressive similarity scores computed on FG model II TTS system for speech synthesis datasets

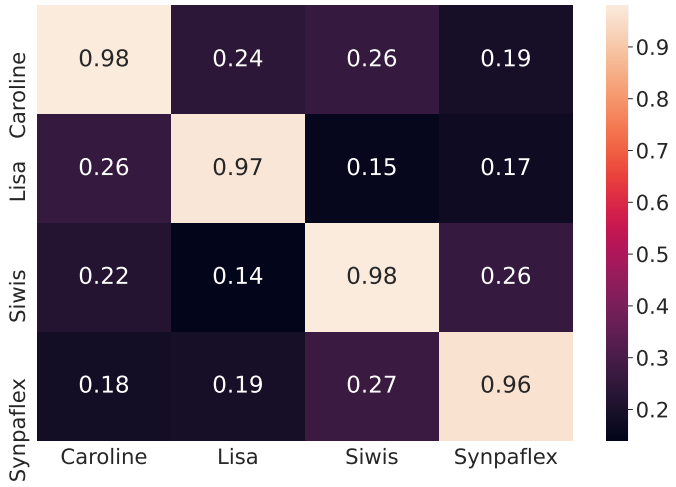

Figure 6.5 – Similarity matrix for speaker similarity scores computed on FG model II TTS system for speech synthesis datasets

representation distinctively.

Expressivity transfer

This section describes the results obtained for the expressivity transfer task. For speaker MOS and expressive MOS, each listener scored five stimuli for each speaker-emotion pair randomly chosen from the test set to evaluate expressivity transfer. Each listener rated three sets of stimuli for each target speaker-emotion combination to compute the speaker MOS and expressivity MOS to estimate expressivity performance. The linguistic contents of the speech stimuli and reference stimuli are not the same during the

Table 6.2 – Evaluation metrics computed to measure the performance of expressivity transfer in parallel setting.

Model	Speaker MOS	Expressivity MOS	Speaker similarity	Expressivity similarity
E2E GST	2.57 ± 0.2	3.05 ± 0.2	0.62	0.53
E2E VAE	2.71 ± 0.3	3.12 ± 0.2	0.69	0.55
E2E GST N-pair	2.65 ± 0.2	3.15 ± 0.4	0.65	0.55
E2E VAE N-pair	**2.90 ± 0.3**	3.33 ± 0.4	**0.71**	0.57
FG model I	2.75 ± 0.3	3.40 ± 0.3	0.68	0.59
FG model II	2.83 ± 0.3	**3.58** ± 0.2	0.68	**0.61**

Table 6.3 – Evaluation metrics computed to measure the performance of expressivity transfer in non-parallel setting.

Model	Speaker similarity	Expressivity similarity
FG model I	0.83	**0.21**
FG model II	**0.84**	**0.21**

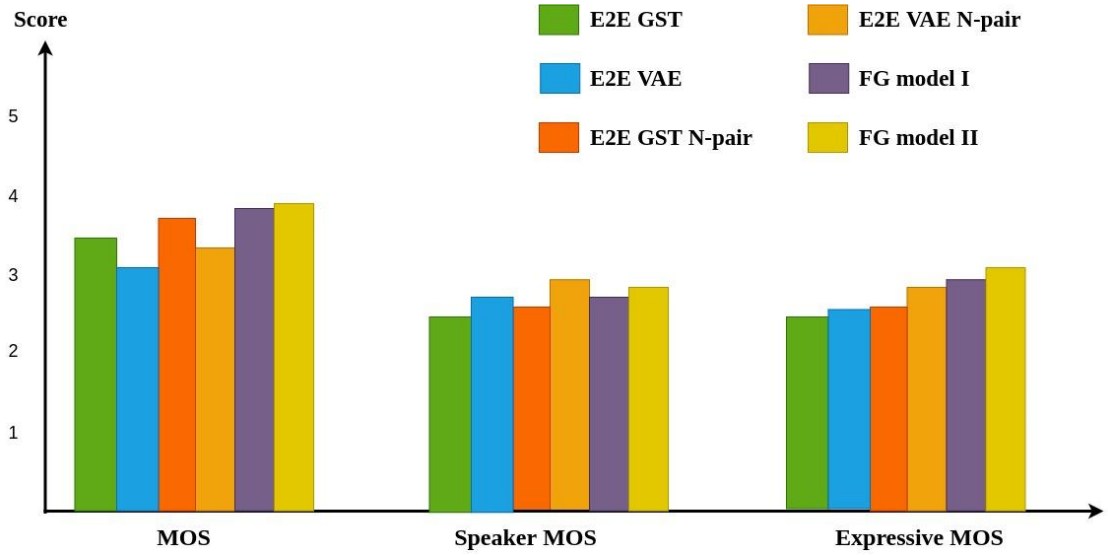

Figure 6.6 – Bar plots for results obtained from subjective listening tests on autoregressive TTS systems

evaluation. We used the reference stimuli from the speech synthesis dataset.

We evaluate the performance of expressivity transfer in a parallel transfer setting, as

stated in Table 6.2. Overall, fine-grained TTS systems outperformed all the other TTS systems based on speaker MOS, expressive MOS, and cosine similarity measures. For ease in the interpretation of results on expressivity transfer in a parallel setting, we show plots for subjective evaluation in Figure 6.6. More particularly, the FG model II system implies that the concatenation of expressivity attention weights and text encoders is critical in extracting local expressivity representation. According to Table 6.2, adding metric learning increases expressivity transfer performance on both expressivity encoders, GST and VAE. E2E VAE N-pair system is the best coarse-grained TTS system for expressivity transfer. Thus, E2E VAE N-pair can generalize a better expressivity latent space as an expressivity encoder. Compared to style tokens, latent variables from the normal distribution provide a more robust normalized representation for the VAE-based expressivity encoder.

For the coarse-grained TTS system, we used pre-computed means of expressive embeddings. Therefore, we only provided results on fine-grained TTS systems in Table 6.3. The obtained results in a non-parallel setting suggest degradation in observed expressivity with high speaker similarity. Therefore, due to degraded performance in Table 6.3, we decided not to conduct the subjective evaluation in a non-parallel transfer setting.

6.3 Non-autoregressive TTS

Since the introduction of Parallel Tacotron, Fastspeech, and Parallel ParaNet, non-autoregressive TTS approaches have enabled faster inference speed. Contrary to the frame-by-frame prediction of Mel spectrogram, non-autoregressive systems generate Mel spectrogram concurrently, thus avoiding the error propagation through previously predicted frames.

There has been limited research conducted on expressivity in non-autoregressive TTS [Lee et al., 2021, Lee, 2021, Shah et al., 2021]. In this section, we present expressivity encoder extension to non-autoregressive TTS systems based on deep generative models, Generative Flow (Glow), and Diffusion probabilistic models (DPM) [Popov et al., 2021, Kim et al., 2020]. The usage of Glow and DPM provides exact log-likelihood estimates, thus providing flexible sampling, inference speed, and high-quality synthesis. These two non-autoregressive TTS systems are trained to optimize the log-likelihood of the Mel spectrogram with hard monotonic alignments. We have described the explanation of Glow-TTS (refer to Section 2.8.2) and Grad-TTS (refer to Section 2.8.3) in Chapter 2. We present extensions to these TTS systems for developing multispeaker expressive non-autoregressive TTS systems. Also, We discussed the deep learning aspects of decoder networks in Section 2.3.

6.3.1 System Components

This section describes various system components used for developing non-autoregressive TTS systems. We used a text encoder, a duration predictor, and a decoder from Grad-TTS and Glow-TTS to develop proposed non-autoregressive TTS systems. We propose modifying a text encoder and adding a speaker encoder and an expressivity encoder to our proposed TTS systems to enable a multispeaker expressive TTS configuration. The

embedding acquired from the expressivity encoder z_e, and the speaker encoder z_s are lengthwise concatenated to the text embedding. By doing so, the single speaker TTS system is adapted to the multispeaker expressive setting. We used two variants of decoder networks, namely Generative Flow and Diffusion probabilistic models. During the training of the decoder network, change in distributions from input to output is learned from the provided training data of phoneme sequence and Mel spectrogram. The Glow decoder is conditioned with z_e and z_s in the affine coupling layer as additional channel inputs. For the DPM decoder, z_e and z_s are given to the decoder as additional channels along with the output from the encoder network. We use a monotonic alignment search based on the Viterbi method [Rabiner, 1989] to find the alignment between phoneme sequence and Mel spectrogram, which is also conditioned on speaker embedding and expressivity embedding.

Speaker encoder

For enabling a multispeaker setting, we created a speaker embedding by providing a one-hot representation to the speaker encoder network. We assigned each speaker identity a unique integer value, which is then converted to on-hot embedding. We used nn.Embedding() from PyTorch library [13] to implement speaker encoder network. The speaker encoder contains a tensor dimension defining the number of speakers and embedding dimension. The speaker encoder is implemented as an embedding matrix, which is a differentiable unit used to find the embedding vector for the given input speaker identity. Initially, the embedding matrix is randomly initialized and learns the embedding similarity during the training phase. We derived z_s of 80 dimensions for conditioning expressive non-autoregressive TTS in a multispeaker setting.

Expressivity encoder

We implemented three techniques to create expressivity embedding. In the first technique, we assigned an integer value to each expressivity class and derived expressivity embedding similar to speaker embedding. This expressivity class identity based expressivity encoder is termed one-hot embedding.

The second technique involves providing reference Mel spectrogram to expressivity encoder, with a model implementation similar to the one explained in Section 4.2.2 (which is based on convolutional recurrent neural network (CRNN) and statistical pooling layer). We implemented an expressivity encoder for reference Mel spectrogram with one significant difference of replacing statistical pooling with self-attentive statistical pooling (SASP) [Okabe et al., 2018]. We termed the reference Mel spectrogram based expressivity encoder as CRNN SASP. In statistical pooling, frame-level features are used to compute mean and standard deviations statistics. Using self-attentive statistical pooling allows assigning different attention weights to different frames and generating weighted means and weighted standard deviations. Thus, self-attention allows deriving long-term variations in expressivity more efficiently.

13. https://pytorch.org/

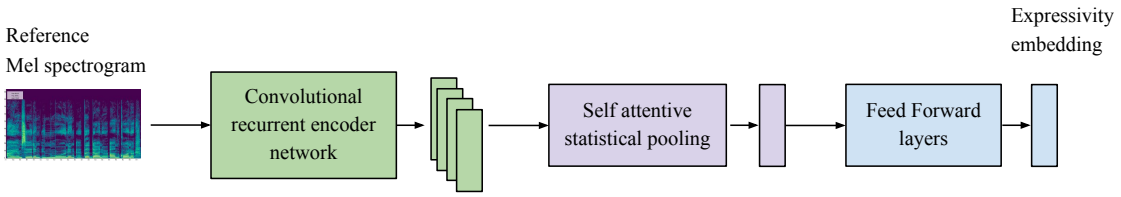

Figure 6.7 – Convolutional recurrent neural network with self-attentive statistical attentive pooling to create expressivity embedding for provided input reference Mel spectrogram.

The CRNN SASP network comprises a convolutional recurrent neural network (CRNN), a self-attentive statistical pooling (SASP), and two feedforward layers with Scaled Exponential Linear Units (SeLU) [Klambauer et al., 2017] applied as activation function, as shown in Figure 6.7. The CRNN is constructed using a single 2-D convolutional layer with single-channel input-output, kernel size of 3x3, along with a single BLSTM layer of 256 hidden units. The usage of the CRNN network assists the network in extracting temporal as well as contextual information from reference Mel spectrogram as frame-level features. In [Okabe et al., 2018], attentive statistical pooling has been proposed in the context of speaker recognition task for improving discriminative distribution across speaker classes. For expressivity representation, we propose to replace classical attention mechanism with self attention [Okabe et al., 2018, India et al., 2019, Zhu et al., 2018b]. This allows the expressivity encoder to extract segmental characteristics from frames that convey expressivity class information as variations in prosodic parameters. The weighted mean and weighted standard deviation are concatenated together as an output of the SASP layer. After that, it is given to two feedforward layers with SeLU activation in between to avoid the internal covariant shift and ensure expressivity embeddings converge towards zero mean and unit variance [Klambauer et al., 2017]. We used 1024 and 512 as hidden units to implement feedforward layers, which are mapped to 80-dimensional expressivity embedding z_e as an output of the last feedforward layer.

The third technique uses a set of traditional emotion features from emotion recognition systems [Arias-Vergara et al., 2017, Vásquez-Correa et al., 2018, Orozco-Arroyave et al., 2018, Dehak et al., 2007] as an input to a multiscale DNN [Peng et al., 2021]. The multiscale expressivity encoder was designed to encompass a variety of traditional emotional features of different frame lengths. We termed this expressivity encoder as multiscale CRNN SASP. We exclusively leveraged traditional emotional features to implement an expressivity encoder for non-autoregressive expressive TTS systems.

Recently multiscale prosody encoders are proposed in addition to Mel spectrogram as input by providing pitch and energy as additional knowledge to learn the intrinsic properties of expressiveness [Zaïdi et al., 2021, Lee et al., 2021, Li et al., 2021b]. We extended the CRNN SASP network to multiscale architecture, which takes into account the fusion of four sets of emotional features. The name multiscale refers to various scales of frame length used in extracting acoustic emotional features from speech. We incorporated four sets of emotional acoustic features for describing articulation, prosody, phonation, and WORLD vocoder features. We described the information regarding emotional acoustic

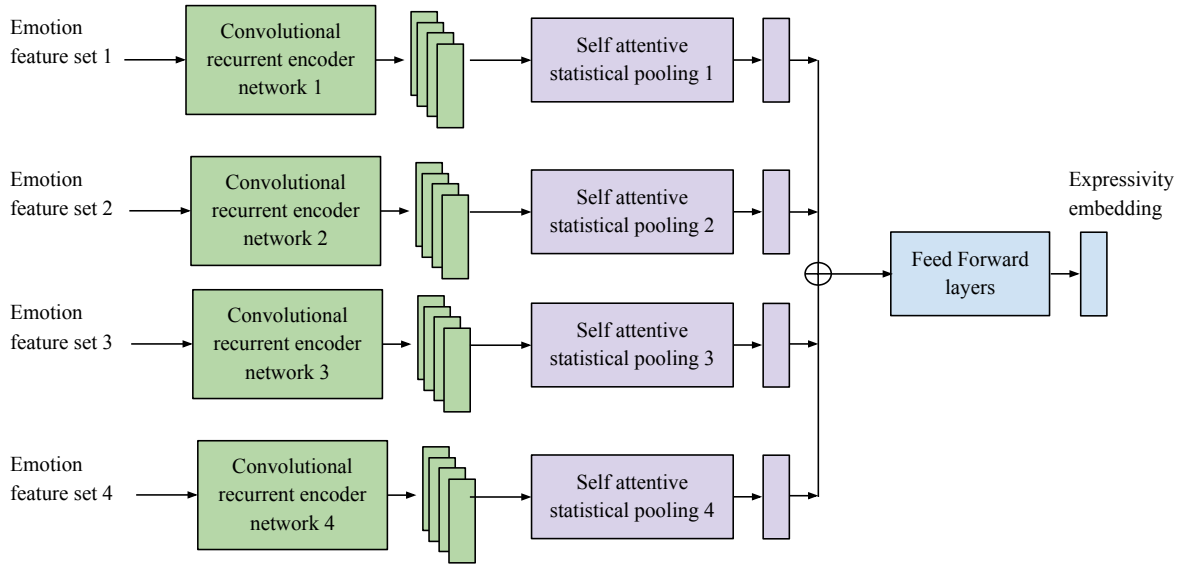

Figure 6.8 – Multiscale expressivity encoder for creating expressivity embedding based on Mel spectrogram as input features

features in Annex A.

We provided each set of emotional features to the CRNN network followed by self-attentive statistical pooling, as shown in Figure 6.8. After that, the output of each CRNN SASP network is concatenated together. We provided combined representation to two feedforward layers with SeLU activation function to obtain the expressivity embedding, z_e. The usage of explicitly providing various emotional features assists the expressivity encoder in learning better variations across the frames and underlining the principle factors in defining expressivity classes. We used the same model parameters as the CRNN network on each scale of the multiscale CRNN SASP network.

Encoder

The encoder network is comprised of a text encoder, duration predictor, and monotonic alignment search algorithm, for which we used the same implementation as stated in Glow-TTS [Kim et al., 2020] and Grad-TTS [Popov et al., 2021]. A detailed explanation of the encoder is described in Section 2.7. Text encoder is composed of transformer network with similar implementation as used in Transformer TTS [Li et al., 2019] with the exclusion of positional encoding. We provide a sequence of phonemes as input to the text encoder. The duration predictor is composed of two convolutional layers, followed by a projection layer that predicts the logarithm of phone duration. We used the same hyperparameters for defining the architecture of a text encoder and the duration predictor module as used in [Kim et al., 2020].

We propose to condition the text encoder and duration predictor on conditional embedding, z_c. The z_c is obtained either by concatenating speaker embedding z_s, and expressivity embedding z_e (refer to Figure 6.9) or only expressivity embedding, z_e (refer to

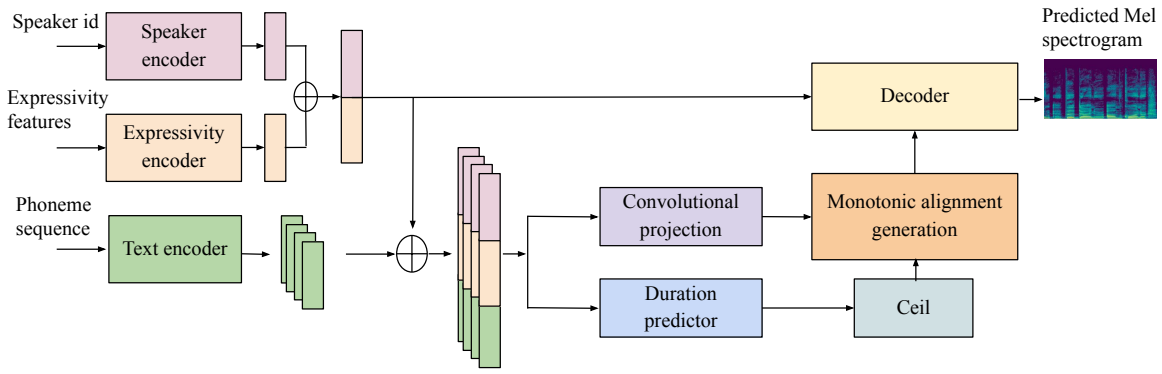

Figure 6.9 – Non-autoregressive expressive TTS architecture by explicitly conditioning text encoder and duration predictor on expressivity embedding and speaker embedding

Figure 6.10). As a result, we investigate how conditional embedding is constructed using a combination of speaker and expressivity information or just expressivity. This investigation provides insight into an assumption of Ekman's emotion model that expressivity patterns are independent of the speaker's voice characteristics. The output of text embedding, z_t is concatenated with z_c as encoder output z_{enc}, Equation 6.1, where $\oplus$ denotes concatenation. After that, z_{enc} is provided as input to the duration predictor.

$$z_{enc} = z_t \oplus z_c \tag{6.1}$$

The primary purpose of the monotonic alignment search is to find the most likely hard alignment between latent variables and prior distribution statistics taken from the encoder output for generating the output Mel spectrogram [Kim et al., 2020]. The use of monotonic hard alignment assures monotonicity and surjectivity, ensuring that spoken text is synthesized in the correct order without skipping any input phoneme sequence.

Therefore, the encoder's output z_{enc} over the sequence of phonemes is used to estimate the statistics, μ_{enc}, and σ_{enc}, as explained in Section 2.8.2. A monotonic alignment search is used during the training phase, and a backpropagation error is passed. Hence, it helps to align prior distribution with the output of the text encoder and duration predictor. Latent variables are sampled from the prior distribution in the inference phase [Casanova et al., 2021].

Decoder: Generative Flow

We discussed the theoretical details regarding Generative Flow (Glow) as a deep generative model in Section 2.3.2. We used Glow as a decoder network in a non-autoregressive TTS system, which is conditioned on encoder output comprising information about the text, speaker, and expressivity.

Similar to Glow-TTS, the Glow decoder is conditioned on a sequence of latent variables generated by a monotonic alignment and output of the encoder network, z_{enc}. As the z_{enc}

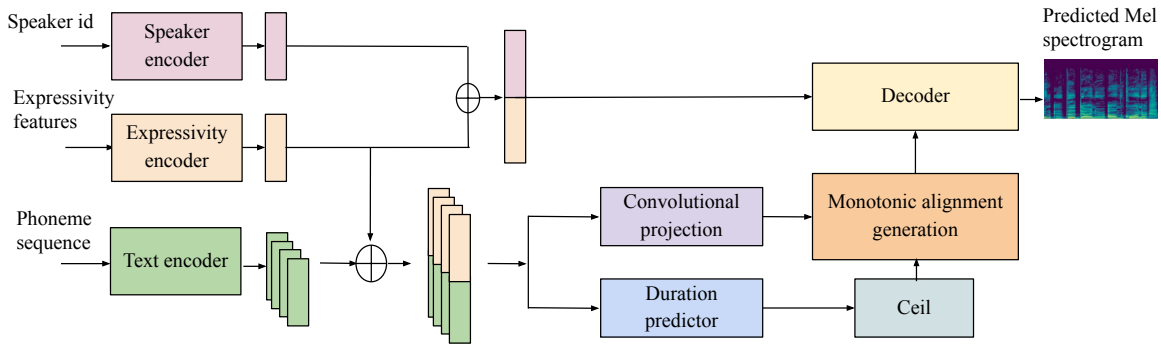

Figure 6.10 – Non-autoregressive expressive TTS architecture by explicitly conditioning text encoder and duration predictor on expressivity embedding

is a concatenation of output of text encoder and conditional embeddings, generated alignment and parameters learned through duration predictor incorporate the knowledge about speaker and expressivity. In the training phase, target Mel-spectrograms are transformed into a sequence of latent variables for negative log-likelihood estimation. Consequently, these latent variables are used for computing alignment search with respect to the output of the encoder network. We propose to provide conditional embedding, z_c, as additional information to affine the coupling layer in each Glow transformation. The loss criterion for training Glow decoder based non-autoregressive TTS is a sum of duration loss and negative log-likelihood estimation, as given below,

$$\mathcal{L}oss = \mathcal{L}_{duration} + \mathcal{L}_{log-likelihood} \tag{6.2}$$

We used the duration loss defined in [Kim et al., 2020], which is calculated using Mean Square Error (MSE) in the logarithmic domain on the duration predictor (DP) in Equation 6.3. Intuitively, $\mathcal{L}_{duration}$ assists the duration predictor in generating duration information in the synthesis phase. Hence, output predicted by DP tries to match the performance of hard alignment obtained by monotonic alignment search in the multispeaker setting, as stated below,

$$\mathcal{L}_{duration} = MSE(DP(sg[z_{enc}]), d_{optimal}) \tag{6.3}$$

where z_{enc}, is encoder output from text encoder, $d_{optimal}$ is optimal durations computed by monotonic alignment, and $sg[.]$ refers to the stop gradient applied on z_{enc}. Thus, $\mathcal{L}_{duration}$ only influences the duration predictor module during backpropagation. We computed $\mathcal{L}_{log-likelihood}$ in the same way as explained in Section 2.8.2. During the synthesis phase, the prior distribution is transformed into Mel spectrogram distribution using cascaded invertible Glow transformation, provided z_c as additional input to affine coupling layer. The prior distribution and alignment are estimated using outputs of the text encoder and duration predictor.

Decoder: Diffusion probabilistic models

The diffusion probabilistic model (DPM) consists of the forward diffusion process and reverse diffusion process. In the forward diffusion process, output data is deconstructed till some simple distribution like Gaussian distribution with zero mean and unit variance is achieved. Conversely, the reverse diffusion process works towards constructing the output distribution and learning the trajectories of the reverse-time forward diffusion by a parameterized neural network, s_θ in the DPM framework [Nichol and Dhariwal, 2021, Ho et al., 2020, Popov et al., 2021]. We explained the DPM as a deep generative model in Section 2.3.3. This section extends the approach detailed in Grad-TTS (detailed in Section 2.8.3) to include a multispeaker expressivity for expressivity transfer.

In the case of a non-autoregressive TTS system that relied on the DPM decoder, the monotonic hard alignment is used to map encoder output, z_{enc} to $\mu_{enc,1:F}$ as aligned latent representation to target Mel spectrogram of F number of frames. The aligned latent representation, $\mu_{enc,1:F}$ is provided as input to the DPM decoder along with conditional embedding z_c, and target Mel spectrogram, X_{Mel}. The parametrized neural network, $s_\theta(X_{Mel}, \mu_{enc,1:F}, z_c, t)$ is implemented for diffusion processing with parameter θ to describe ordinary differential equation (ODE), where $t \in [0, T]$, and terminal time, T. During the inference, the time horizon is selected empirically, which controls the inference speed and quality of synthesized speech.

For conditioning speaker and expressivity information in DPM decoder, we provided X_{Mel},$\mu_{enc,1:F}$,z_c, and t as channel-wise input to the U-net module for diffusion processing. The transformed and aligned latent representation, μ_{enc} is used in defining the terminal distribution $X_{terminal} \sim \mathcal{N}(\mu_{enc}, \tau^{-1}I)$. The τ is the temperature parameter that aids in controlling the quality of output Mel spectrograms in reverse diffusions. Noise schedule β_t and time $t = 0 : T$, where T is time horizon predefined parameter to conduct cascaded forward diffusions and reverse diffusions. The DPM decoder network based on s_θ is implemented using U-net architecture as designed in [Ronneberger et al., 2015, Ho et al., 2020], for more details of implementation refer Section 2.3.3. Furthermore, we used the same hyperparameters as the original Grad-TTS system, with addition of 2 more channels for 80 dimensional expressivity embedding and 80 dimensional speaker embedding.

$$dX_t = \frac{1}{2}(\mu_{enc,1:F} - X_{Mel} - s_\theta(X_{Mel}, \mu_{enc}, z_c, t))\beta_t dt \tag{6.4}$$

The duration predictor provides the required number of frames for generating the Mel spectrogram during the synthesis phase. Afterward, the reverse diffusion process evolves as described by Equation 6.4. Euler scheme is used for solving ODE backward in time for parameter θ [Chen et al., 2018, Song et al., 2021].

Three loss terms, duration loss, encoder loss, and diffusion loss, describe the loss criterion for training the TTS system based on the DPM decoder, as stated in Equation 6.5. The duration loss is the same as the one used in the Glow decoder. The encoder loss, $\mathcal{L}_{encoder}$ is based on negative log-likelihood. Since the DPM framework starts decoding random noise with a normal distribution $\mathcal{N}(\mu_{enc}, I)$, it is easier for the DPM decoder to sample noise that is already close to the output distribution. As a result, it is advised to include $\mathcal{L}_{encoder}$ as one of the auxiliary loss terms. In [Popov et al., 2021] diffusion loss

$\mathcal{L}_{diffusion}$ is described as the expectation of weighted losses associated with estimating gradients of log-density of noisy data at different time horizons, T, starting from $t = 0$.

$$\mathcal{L}oss = \mathcal{L}_{duration} + \mathcal{L}_{encoder} + \mathcal{L}_{diffusion} \tag{6.5}$$

6.3.2 Expressivity Transfer

For transfer of expressivity, we considered two cases similar to fine-grained end-to-end expressive TTS models, parallel transfer and non-parallel transfer. In the case of parallel expressivity transfer, we provided a suitable expressivity feature as input to the expressivity encoder. The expressivity feature has the same textual content as the input phoneme sequence given to the text encoder.

On the other hand, non-parallel expressivity transfer involves a mismatch of textual content between the input given to the expressivity encoder and the input given to the text encoder. For non-parallel transfer, we selected reference acoustic features extracted from longer speech utterances of desired expressivity. In this work, we generated results on both types of expressivity transfer techniques.

6.3.3 Experimental Setup

Table 6.4 – Experimentation setup with non-autoregressive (Non-AR) TTS using two decoder networks, Glow and DPM and three architectures for expressivity encoder, One-hot embedding, CRNN SASP, and Multiscale CRNN SASP. Also, we presented various architecture by conditioning text encoder and duration predictor either on expressive embedding z_e, and speaker embedding z_s or on expressive embedding z_e.

Non-AR TTS Model	Expressvity encoder architecture	Text encoder and duration predictor conditioned on	Decoder
Glow base	pretrained	z_e, z_s	Glow
Glow-one-hot-embedding	One-hot embedding	z_e, z_s	Glow
Glow-attention-pool	CRNN SASP	z_e, z_s	Glow
Glow-attention-pool*	CRNN SASP	z_e	Glow
Glow-multiscale	Multiscale CRNN SASP	z_e, z_s	Glow
DPM-one-hot-embedding	One-hot embedding	z_e, z_s	DPM
DPM-attention-pool	CRNN SASP	z_e, z_s	DPM
DPM-attention-pool*	CRNN SASP	z_e	DPM
DPM-multiscale	Multiscale CRNN SASP	z_e, z_s	DPM

Initially, we conducted experiments using an expressivity encoder based on the CRNN statistical pooling network, for which we provided input as reference Mel spectrogram

of desired expressivity. We used four French female speech datasets for this experiment: Caroline expressive, Lisa, Siwis, and Synpaflex. We trained a Glow decoder-based non-autoregressive TTS system using a CRNN statistical pooling network as an expressivity encoder. We termed this system as Glow base, also stated in Figure 6.3.3. The obtained results with Glow base TTS system was neither able to synthesize comprehensible expressive speech nor transfer expressivity due to an uneven data distribution across neutral and other expressivity classes. Furthermore, the time required for developing such a model approximately takes more than 200hrs training on the GPU resources (GetForce RTX 2080 Ti, GeForce GTX 1080 Ti, Tesla T4) available on the Grid5000 computing node.

Following this observation, we opted for only two speakers, Siwis neutral dataset and Caroline expressive dataset, to develop non-autoregressive expressive TTS systems. Furthermore, we selected only three distinctive expressivity classes, anger, sadness, and joy, out of six expressivity classes excluding neutral voice. After analyzing Ekman's emotion model described in Section 1.2, these three expressivity classes are selected, showing significant distinguishing characteristics of expressivity. Also, using two speech datasets and a limited number of expressivity classes allows us to train non-autoregressive systems more quickly than with all the French speech synthesis datasets. We used 5hrs of neutral speech data for Siwis, while for Caroline expressive data, we used approximately 1hr of speech data of anger, sadness, and joy as expressivity classes. Our main objective is to transfer expressivity to the speaker's voice for which no expressive speech is available. We decided not to use Caroline neutral data; the aim is to transfer expressive attributes to Siwis speaker's voice. Each speech corpus is split into train, validation, and test sets in $80 : 10 : 10$ ratio, respectively.

The model components explained in Section 6.3.1 are used to develop various TTS models for expressivity transfer tasks in a non-autoregressive framework. For observing decoding capabilities of Flow models versus probabilistic diffusion models, we investigated two decoder networks, namely, Glow and DPM. Afterward, we explored three architectures for expressivity representation based on three kinds of input to expressivity encoder, one-hot embedding, Mel spectrogram, and sets of classical emotional features used in emotion recognition systems. For expressive speech synthesis, variations in phoneme duration play a crucial role [Kulkarni et al., 2020c] in perceived expressivity classes. Based on previously proposed emotion models, variations across the expressivity classes are supposed to be speaker-independent. Therefore, we analyzed this assumption by conditioning text encoder and duration predictor only on expressivity embedding, as illustrated by Figure 6.10.

This thesis work proposes eight different non-autoregressive TTS models for expressivity transfer based on variations in decoder architecture, expressivity encoder, and conditioning of encoder network, as stated in Table 6.3.3. We followed the same hyperparameters, as well as the training procedures explained in [Kim et al., 2020, Popov et al., 2021]. All the non-autoregressive TTS systems were trained for 1500 epochs, including Glow and DPM architectures. For non-autoregressive TTS models, we have used a sampling rate of 22050 Hz and Mel spectrograms as acoustic features to be predicted by the end-to-end TTS system. Recently proposed HiFi-GAN neural vocoder shown better results for speech waveform generation than WaveGlow [Kong et al., 2020]. Therefore, we replaced WaveGlow based neural vocoder with HiFi-GAN. We used a pretrained HiFi-GAN model

trained on the LJS dataset released on the official implementation website [14].

6.3.4 Results

We used the objective evaluation metrics explained in Section 4.2 to measure the performance of proposed non-autoregressive TTS systems. We excluded the Glow base system for subjective evaluation due to the overall degraded quality of synthesized speech by Glow base model.

We conducted a subjective evaluation using perceptive listening tests described in Section 4.3 for evaluation of TTS systems as well as expressivity transfer. The speaker MOS, expressive MOS, speaker cosine similarity score, and expressive cosine similarity scores are used to evaluate the performance of the presented architectures for the transfer of expressivity onto target speaker voices. The MOS, speaker MOS, expressive MOS are computed with associated 95% confidence intervals.

TTS system

A total of 20 French listeners participated in this MOS test to evaluate eight non-autoregressive TTS systems. We selected 48 speech stimuli from eight non-autoregressive TTS systems and four speech stimuli from speech synthesis datasets. The main purpose of having a subset of original speech stimuli is to validate the scores provided by listeners. Furthermore, we created a global test of 176 speech stimuli with unique textual content. During each MOS listening test, speech stimuli are randomly shuffled across the listening test.

Table 6.5 – TTS Evaluation metrics computed to measure the performance of non-autoregressive end-to-end TTS systems

Model	MOS	MCD	BAP	Lf0 RMSE	Speaker similarity	Expressivity similarity
Glow base	—	4.59	0.9982	18.40	0.79	0.94
Glow-one-hot-embedding	3.63 ± 0.1	4.57	**1.11**	18.80	0.79	0.81
Glow-attention-pool	3.72 ± 0.2	4.56	1.13	18.65	**0.81**	0.79
Glow-attention-pool*	3.57 ± 0.2	**4.51**	1.20	18.51	0.75	0.75
Glow-multiscale	3.86 ± 0.1	4.53	1.15	**18.24**	0.80	0.76
DPM-one-hot-embedding	3.68 ± 0.1	4.61	1.20	18.94	0.77	0.82
DPM-attention-pool	3.85 ± 0.2	4.54	1.24	18.85	0.79	**0.83**
DPM-attention-pool*	3.55 ± 0.2	4.53	1.23	18.47	0.48	0.43
DPM-multiscale	**3.94 ± 0.2**	**4.51**	1.27	18.88	0.71	0.78

Table 6.5 describes the performance of non-autoregressive TTS systems for speech synthesis tasks. The obtained results on subjective and objective evaluation metrics are

14. https://github.com/jik876/hifi-gan

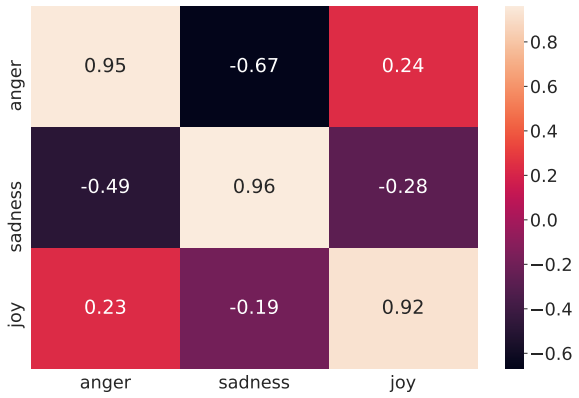

Figure 6.11 – Matrix representing expressive similarity score computed on joy, anger, and surprise expressivity classes on Glow-multiscale TTS system.

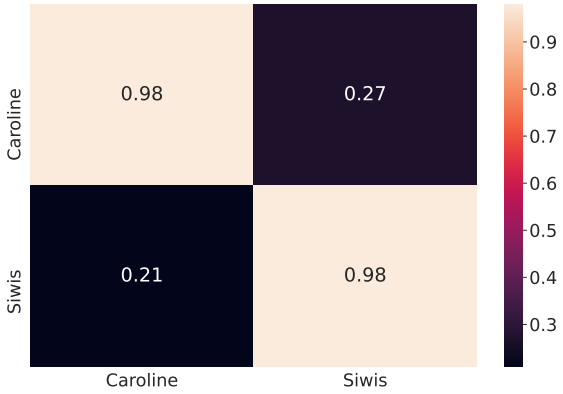

Figure 6.12 – Matrix for speaker similarity score computed to analyze similarity between Siwis and Caroline speaker's voice using Glow-multiscale TTS system.

consistent with each other. From Table 6.5, the usage of reference acoustic features to expressivity encoder improves the quality of synthesized speech instead of using one-hot-embedding as expressivity representation. Also, using a multiscale expressivity encoder provides an improvement in synthesizing speech for the TTS system based on Glow, as well as on DPM. This suggests the significance of using multiple acoustic features, providing meaningful expressivity representation.

Furthermore, the DPM-multiscale system obtained the highest MOS score. But the Glow-multiscale system illustrated better expressivity and speaker scores than the DPM-multiscale system. Glow-one-hot-embedding system and DPM-one-hot-embedding better expressivity similarity scores but at the cost of intelligibility performance degradation suggested by MOS scores. Even though the DPM-one-hot-embedding system scored better expressivity similarity scores, it degrades intelligibility performance. The conditioning of the encoder only with expressivity embedding leads to poor performance on TTS evaluation metrics for both Glow-attention-pool* system and DPM-attention-pool* system. We also observed that DPM-based expressive TTS systems synthesized speech with better expressivity similarity scores than the Glow-based expressive TTS systems.

We also plotted the matrix representing the similarity between expressivity scores for three expressivity classes, joy, anger, and surprise in Figure 6.11. Furthermore, for speaker similarity scores on two speaker's voices, Siwis and Lisa expressivity similarity score and speaker similarity scores in Figure 6.12. From Figure 6.11, joy and anger expressivity classes are closer to each other than sadness. Additionally, the matrix in Figure 6.12 provides the usage of only two speakers in a multispeaker setting creating a distinctive speaker representation compared to the autoregressive TTS system.

Expressivity transfer

A subjective evaluation setting is designed to get speaker MOS and expressive MOS from the perceptive listening test for expressivity transfer. We conducted a subjective listening test with 20 native French participants. We provided two sets of speech stimuli for each expressivity class; each set comprises four speech stimuli from four different non-autoregressive TTS systems, in a total of 48 samples and 4 speech samples from original speech samples.

To validate the correctness of scores submitted by listeners, we also provided speech stimuli from the E2E GST system and speech synthesis datasets. We selected scores from listeners consistent with results on speech stimuli from E2E GST and speech synthesis datasets. The reference speech stimuli for speaker and expressivity were chosen from speech synthesis datasets to measure speaker MOS and expressive MOS.

Table 6.6 – Evaluation metrics computed to measure the performance of expressivity transfer in parallel setting

Model	Speaker MOS	Expressive MOS	Speaker similarity	Expressivity similarity
Glow base	—	—	0.73	0.31
Glow-one-hot-embedding	3.12 ± 0.1	2.81 ± 0.2	0.68	0.35
Glow-attention-pool	3.20 ± 0.1	2.73 ± 0.1	0.70	0.31
Glow-attention-pool*	2.86 ± 0.2	2.91 ± 0.2	0.52	**0.42**
Glow-multiscale	**3.34 ± 0.2**	2.87 ± 0.1	**0.73**	0.30
DPM-one-hot-embedding	3.13 ± 0.3	2.78 ± 0.2	0.68	0.35
DPM-attention-pool	3.06 ± 0.2	2.73 ± 0.1	0.62	0.32
DPM-attention-pool*	2.97 ± 0.1	**2.95 ± 0.3**	0.67	0.36
DPM-multiscale	3.15 ± 0.1	2.81 ± 0.2	0.69	0.34

In each listening test, speech stimuli are randomly selected from a more extensive test set with unique textual content. We opted for speaker similarity score, and expressivity similarity score for objective evaluation of expressivity transfer.

Table 6.6 describes the performance of expressivity transfer in a parallel transfer setting with subjective evaluation metrics and objective evaluation metrics. From Table 6.6, speaker MOS expressive MOS scores are inline with the objective measures of speaker similarity and expressivity similarity.

Table 6.7 – Evaluation metrics computed to measure the performance of expressivity transfer in non-parallel setting.

Model	Speaker similarity	Expressivity similarity
Glow base	0.80	0.11
Glow-one-hot-embedding	0.75	**0.35**
Glow-attention-pool	0.76	0.30
Glow-attention-pool*	0.65	0.31
Glow-multiscale	**0.79**	0.29
DPM-one-hot-embedding	0.65	0.30
DPM-attention-pool	0.72	0.30
DPM-attention-pool*	0.66	0.29
DPM-multiscale	0.77	0.22

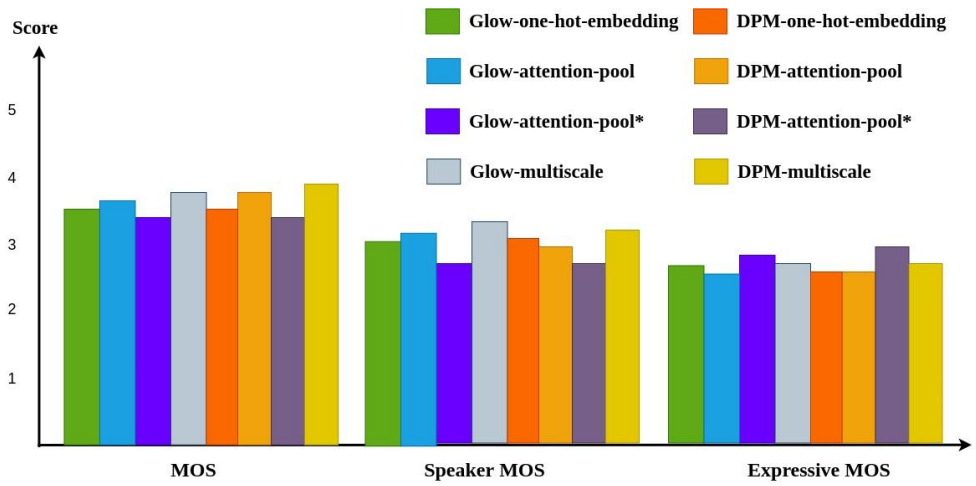

Figure 6.13 – Bar plots for results obtained from subjective listening tests on non-autoregressive TTS systems

The highest expressivity scores were obtained by the Glow-attention-pool* and DPM-attention-pool; hence, conditioning the encoder network exclusively with expressivity embedding increases the expressivity information in synthesized speech. But it also results in lower speaker MOS. From Figure 6.13, speaker MOS, expressive MOS, and MOS scores clearly indicate that usage of CRNN SASP based expressivity encoder with multiscale as well as with Mel spectrogram improves the overall performance in expressivity transfer as well as for intelligibility.

For both decoders, the multiscale expressivity encoder demonstrates the ability to transfer expressivity than Glow-attention-pool and DPM-attention-pool systems. From Table 6.7, for non-parallel transfer, the Glow-multiscale and DPM-multiscale systems

showed the best scores on cosine similarity for speaker and expressivity.

6.4 Discussion

In coarse-grained approaches, we proposed to use deep metric learning, which improves the overall performance of coarse-grained TTS systems. We opted for multiclass N-pair loss to consider multiple negative samples and a single positive sample, thus inter-relation between expressivity classes is taken into account. For transfer of expressivity VAE based expressivity encoder generalizes emotion representation better than GST.

We proposed using fine-grained TTS architectures in autoregressive TTS settings, where information from the expressivity encoder and text encoder is used to construct expressive attention weights. These attention weights take into account positional phoneme information and its correlation with prosodic information from hidden representation from the expressivity encoder. Our experimentation with fine-grained systems suggests that concatenation of output of text encoder and expressivity attention weights plays a vital role in improving overall performance in synthesizing speech, thus resulting in better intelligibility than other end-to-end TTS systems. We also observed that in a non-parallel transfer setting, autoregressive TTS systems could synthesize speech with lower expressivity.

In non-autoregressive TTS systems, the obtained results indicated that multiscale architecture of expressivity encoder with provided input containing acoustic features ranging from the pitch, aperiodicity, Mel spectrogram, energy, jitter, shimmer, etc. Therefore, the multiscale expressivity encoder creates the embedding containing information associated with various prosodic factors.

In a non-parallel setting, the performance of all the non-autoregressive TTS systems was closer to results from parallel transfer settings. In non-autoregressive TTS systems, we used a monotonic alignment search for estimating the most likely hidden alignment without any external aligner. The usage of adding speech samples from synthesis datasets helped us track the validity of listening tests.

The parametric TTS systems conduct expressivity transfer in acoustic space only, which lacks the interpolation in duration prediction. Thus, the end-to-end TTS system influences the prosody of synthesized speech and the alignment of synthesized speech for each emotion. Contrary to the parametric TTS framework and usage of the acoustic model, the end-to-end system enables the disentanglement over the acoustic feature space and the associated duration paradigm of the expressivity class to be synthesized.

6.5 Conclusion

In this chapter, we presented various end-to-end TTS systems to transfer expressivity in a multispeaker setting. We investigated two frameworks of end-to-end TTS systems, namely autoregressive and non-autoregressive. In the case of the autoregressive framework, coarse-grained and fine-grained expressivity transfers were presented, and obtained results indicate that usage of fine-grained expressivity transfer certainly improves the

performance of expressivity transfer. Furthermore, We explored non-autoregressive TTS systems for expressivity transfer tasks with two kinds of decoder networks, namely Glow and DPM. Additionally, we also explored various types of reference input to the expressivity encoder. The proposed non-autoregressive TTS system demonstrates that the usage of a multiscale expressivity encoder provides a better representation of expressivity to transfer expressivity attributes to the target speaker's voice. We investigated parallel and non-parallel expressivity transfer based on input textual content to expressivity encoder and text encoder.

The autoregressive TTS system based on a fine-grained approach showed MOS scores closer to the non-autoregressive TTS systems for expressivity encoder based on multiscale architecture. In non-autoregressive TTS systems, we experimented with two approaches for conditioning the TTS system on speaker and expressivity embeddings. The results indicated that conditioning only with expressivity embedding leads to better expressivity transfer. But it leads to poor performance for TTS tasks compared to other systems.

Chapter 7

Conclusion

This chapter summarizes the contributions made in the thesis study and suggests several prospective future research directions in this area. The primary objective of the thesis work was to develop an expressive TTS system in a speaker's voice for which only neutral speech data is available. We investigated deep learning approaches for exploring disentanglement of speaker information and expressivity for expressivity transfer to the target speaker's voice.

7.1 Contribution of Thesis

We presented evaluation metrics used to measure the TTS system's performance and expressivity transfer in Chapter 4. For assessing different characteristics of synthesized speech, we noticed that using multiple evaluation metrics helps measure the overall performance of TTS more precisely than using only a single metric. In addition to traditional TTS objective evaluation metrics such as MCD, BAP, and Lf0 RMSE, and cosine similarity to measure the level of expressivity and speaker attributes. Due to the unavailability of the expressive speech samples in the target speaker's voice, we proposed using cosine similarity for measuring the system's efficiency to transfer the expressivity. Therefore, multiple objective metrics enabled knowledge of desired attributes like pitch information from Lf0 RMSE, energy-related information from MCD, etc.

We presented speaker MOS and expressive MOS using perceptive listening tests for subjective evaluation of expressivity transfer. This usage of two separate scores aims at evaluating two different attributes present in speech synthesized using expressivity transfer. Speaker MOS to observe the perceived similarity of speaker characteristics in synthesized expressive speech. And expressive MOS for observing perceived expressivity in synthesized expressive speech. We designed both speaker MOS and expressive MOS listening test setup by modifying the ABX listening test on an absolute ranking scale, where the minimum score is one and the maximum score is five. The obtained results from Chapter 5 and Chapter 6 demonstrate the relatedness of cosine similarity metrics to the speaker MOS and expressive MOS. Therefore, we found it helpful in evaluating new approaches without conducting the time-consuming subjective evaluation process. Furthermore, it is imperative to not provide too many speech stimuli's during subjective

evaluation, thus avoiding the weariness of listeners due to cognitive loads. One challenge encountered during thesis work is a subjective evaluation, which involves finding a sufficient number of native French listeners as participants and the time required for subjective evaluation.

As a first step towards solving expressivity transfer, we developed single speaker single emotion parametric TTS systems for French speakers, serving as baseline systems. In parametric TTS systems, we used BLSTM layers to implement the duration and acoustic models. Furthermore, the analysis showed lower expressivity similarity scores on sadness and disgust compared to other expressivity classes. Therefore, this indicates the closeness of expressivity classes in expressive speech synthesis.

After that, we focused on implementing a multispeaker expressive parametric TTS system. As additional input to TTS, we used speaker embedding and expressivity embedding and then used appropriate embeddings to predict acoustic features for the target speaker and expressivity. We proposed to use the transfer learning technique to extract speaker information from x-vector embeddings and fine-tune further with a feedforward neural network on French speech synthesis datasets. The proposed way of creating speaker embeddings showed that parametric TTS systems obtained higher speaker MOS scores than the autoregressive TTS systems. In the case of end-to-end TTS systems, we used one-hot-embedding representation to create speaker embeddings. We opted out of the usage of Kaldi toolkit-based x-vector embeddings as it was complicated to integrate Kaldi functions inside the end-to-end deep neural network architecture.

As expressivity representation should be disentangled from speaker attributes, it may result in speaker leakage. We presented variational inference to learn the meaningful latent representation of expressivity using variational autoencoders and normalizing Flow in parametric TTS systems.

Furthermore, we also proposed to use deep metric learning-based multiclass N-pair loss criterion along with loss term from variational inference. The obtained results demonstrated that using such adversarial loss criterion leads to better clustering of expressivity class in latent space. As a result, clustered expressivity embeddings provide better estimates of pre-computed means used for expressivity transfer. Due to the tractability of normalizing Flow models as well as the ability to map complex distribution to a simple one, inverse autoregressive Flow with metric learning as the acoustic model showed better results than variational autoencoders.

We dedicated Chapter 6 in explaining and evaluating proposed end-to-end TTS systems in the context of expressivity transfer. First, we detailed the autoregressive TTS systems based on Tacotron 2 architecture. We developed two multispeaker expressive TTS systems using global style token and variational autoencoder as expressivity encoder networks. Both proposed systems created phoneme-independent expressivity embeddings using a coarse-grained autoregressive TTS system. In the same way that we trained parametric TTS systems, we also trained autoregressive TTS systems with multiclass N-pair loss. The results showed an improvement in speech synthesis quality and expressivity transfer.

As speech's expressiveness varies with phonemes' context, we employed the fine-grained framework using a positional attention mechanism to derive hidden phoneme-dependent expressivity representation. Positional attention learns to derive the expressive

attention weights varying with the output of the text encoder. Furthermore, our proposed fine-grained autoregressive TTS system is constructed with multistage attention to generating expressivity representation at the local and global levels. The experimentation with fine-grained TTS systems indicated that concatenation of output of text encoder and expressivity attention weights improve the performance in synthesizing speech, thus resulting in better intelligibility and higher expressivity than other TTS systems. We also analyzed that autoregressive TTS systems could not perform expressivity transfer in non-parallel settings, where the textual content of reference acoustic feature to expressivity encoder is different from the input text to the TTS system.

In addition to autoregressive TTS systems, we also presented expressivity transfer using non-autoregressive TTS systems. We opted for probability density estimators Glow and diffusion probabilistic models as deep generative models to learn the mapping between sequence of latent variable and Mel spectrogram as output. For the expressivity transfer task, we modified the TTS architecture with a speaker encoder and expressivity encoder. The output of these encoders is used for conditioning text encoder and decoder networks.

Comparing the experimentation with two deep generative models suggests that Glow decoders are better at learning the expressive attributes and speaker characteristics than the diffusion probabilistic models. Furthermore, we experimented with three expressivity encoders consuming different features to represent expressivity. These expressivity features include a one-hot representation of expressivity class, a Mel spectrogram of desired expressivity class, and multiple emotional acoustic features derived with different frame lengths. We proposed a convolutional recurrent neural network (CRNN) with self-attention pooling to create expressivity embedding from the Mel spectrogram. The usage of self-attention pooling enables highlighting relevant expressivity attributes from the output of the CRNN network. We further improved this architecture by providing multiple acoustic features containing different frame lengths to multiple blocks of CRNN with self-attention pooling networks. The results obtained on non-autoregressive TTS systems showed that a multiscale expressivity encoder provided a better expressivity encoder for both decoders (Glow and DPM). We also noticed that even though one-hot-embedding-based systems showed higher expressivity scores in cosine similarity, one-hot-embedding systems showed lower MOS scores.

We experimented with analyzing the effect of conditioning encoder either only with expressivity embedding or with both expressivity and speaker embedding. The obtained results show the poor performance as a TTS system when conditioning encoder only with expressivity embedding. The expressivity transfer task showed significant expressivity scores on objective and subjective evaluation metrics. But such conditioning of encoder only with expressivity encoder also leads to speaker leakage reflected by speaker MOS and speaker similarity scores.

The parametric TTS systems conduct expressivity transfer in acoustic space only, which lacks the interpolation in duration prediction. The end-to-end TTS systems incorporate both acoustic and duration knowledge regarding synthesizing speech using single DNN architecture. Thus, the end-to-end TTS system influences the prosody of synthesized speech and the alignment of synthesized speech for each emotion. The investigation of various deep learning architectures and TTS frameworks suggested fine-grained TTS systems showed one of the best performances in parallel transfer settings. Additionally, it

is also observed that results from Chapter 6 suggest that non-autoregressive TTS systems showed lower speaker leakage with lower expressiveness compared to autoregressive TTS systems.

In a non-parallel setting, the performance of all the non-autoregressive TTS systems was closer to results from parallel transfer settings. Due to duration loss and monotonic hard alignment, proposed non-autoregressive TTS systems avoided the pronunciation issues observed with systems like Tacotron 2. Even though non-autoregressive TTS systems were not able to achieve similar performances as fine-grained TTS systems for expressivity transfer, they demonstrated better performance for text-to-speech synthesis task.

7.2 Future Work

The future work can be envisioned with a focus on two aspects: TTS systems' overall performance and model capability of expressivity transfer. The usage of neural network-based grapheme to phoneme converter jointly trained end-to-end TTS systems can improve phonetization errors. Furthermore, it can also assist the end-to-end TTS system in learning variations in expressive speech.

We incorporated neural vocoders pre-train on the English speech synthesis dataset. Therefore, to boost the neural vocoder performance, we also recommend to re-train neural vocoders on French speech synthesis datasets. In coarse-grained autoregressive TTS systems, we used WaveGlow neural vocoder specifically trained on French speech synthesis datasets. For fine-grained autoregressive TTS, we incorporated WaveGlow neural vocoder pretrained on the LJSpeech dataset. The informal listening test between the pretrained model on French speech synthesis datasets and the LJSpeech dataset has not shown any significant difference. Additionally, the training procedure for neural vocoder is a time-consuming process. Nevertheless, fine-tuning neural vocoder on French speech synthesis datasets can assist in further improving the quality of synthesis.

More in-depth research should be required to create a single objective evaluation metric for evaluating the performance of expressivity transfer. Therefore, in the future, research should be conducted to formalize the relation between various evaluations such as cosine similarity, and subjective evaluations such as MOS, speaker MOS, and expressive MOS.

The performance of end-to-end TTS systems heavily relies on a large amount of training data for achieving state-of-the-art results. Thus, it is prudent to use a substantial amount of expressive speech data. Additionally, expressive TTS systems should understand factored representation of speaker attributes and expressivity. Incorporating multiple expressive speech synthesis datasets will allow the system to learn to disentangle speakers and expressivity. Throughout the thesis work, we developed expressive TTS systems without considering the semantics and sentiment of textual content. It will be interesting to analyze the effect of the sentiment of text on expressive speech synthesis.

The proposed fine-grained TTS systems showed the usefulness of phoneme-dependent expressivity. Also, the multiscale expressivity encoder corroborated the learning of multiple prosodic features. In the future, it will be desirable to conduct experimentation with TTS systems with the extension of weighted predicted pitch information to a non-autoregressive decoder. The usage of learnable weights in pitch prediction can control the

pitch for a given expressivity class. Furthermore, to address speaker leakage in expressivity transfer, there is a need to design neural network systems with speaker-independent expressivity embeddings. Using the metric learning loss criterion proposed using a single reference acoustic feature to create two separate embeddings to represent speaker and expressivity. Recently DNN architecture in [Li et al., 2021a] proposed to estimate the loss to ensure that predicted acoustic features contain desired speaker attribute and expressivity classes. It is hard to identify which neural network parameters represent the attributes of speaker and expressivity. However, conducted thesis work demonstrated the usage of expressivity transfer, and it is evident to disentangle the expressivity and speaker characteristics.

Appendix A

Acoustic emotional features

We describe the information regarding acoustic emotional features used in Section 6.3 as an input to a multiscale expressivity encoder. The name multiscale refers to various scales of frame length used in extracting acoustic emotional features from speech. We incorporated four sets of acoustic features for describing articulation, prosody, phonation, and WORLD vocoder. Apart from WORLD vocoder features, we used DisVoice [15] python library to compute features representing articulation, prosody, and phonation.

Acoustic emotional feature set 1: articulation

Articulation features are formed using 58 features computed for a frame length of 40 ms with a time-shift of 20 ms in onset transitions. The onset transitions are the transitions from voiceless-to-voiced [Arias-Vergara et al., 2021]. These features are mainly computed using Bark band energies (BBE), and Mel frequency cepstrum coefficient (MFCC) with derivatives. In BBE features, the Bark scale is used to derive the information corresponding to critical bands of human hearing each having a width of one Bark. By representing spectral energy (in dB) over the Bark scale, a closer correspondence is obtained with spectral information processing in the ear. Articulation features used in acoustic emotional feature set 1 are described below,

1. 1-22, Bark band energies in onset transitions (22 BBE)
2. 23-34, Mel frequency cepstral coefficients in onset transitions (12 MFCC)
3. 35-46, the first derivative of the MFCCs in onset transitions (12 DMFCC)
4. 47-58, the second derivative of the MFCCs in onset transitions (12 DDMFCC)

Acoustic emotional feature set 2: prosody

The prosody features extracted correspond to the Lagrange polynomial modeling the F0 contour, the Lagrange polynomial to model energy contour, and the duration of the voiced segment. We extracted 13 prosody features as detailed in [Dehak et al., 2007]. Prosody features used in acoustic emotional feature set 2 are described below,

1. 1-6, coefficients of 5-degree Lagrange polynomial to model F0 contour

15. https://github.com/jcvasquezc/DisVoice

2. 7-12, coefficients of 5-degree Lagrange polynomial to model energy contour
3. 13, duration of the voiced segment

Acoustic emotional feature set 3: phonation

The phonation feature set includes seven features computed for a frame length of 40 ms. Phonation is usually referred to as the process by which the vocal folds produce certain sounds through quasi-periodic vibration. Phonation features used in acoustic emotional feature set 3 are described below,

1. First derivative of the fundamental Frequency
2. Second derivative of the fundamental Frequency
3. Jitter
4. Shimmer
5. Amplitude perturbation quotient
6. Pitch perturbation quotient
7. Logaritmic Energy

Acoustic emotional feature set 4: WORLD vocoder

The fourth set of acoustic emotional feature sets refers to features extracted using the WORLD vocoder. We described 187 acoustic features in Section 3.4, referring to 180 MGC, 3 Lf0, 3 bap, and 1 voiced-unvoiced flag extracted over a frame interval of 5msec.

Appendix B

Long résumé en français

B.1 Introduction

La parole est l'un des moyens de la communication humaine. Avec l'évolution des technologies, l'idée que les machines, les ordinateurs et les robots puissent parler et s'exprimer comme des êtres humains est devenue une réalité avec la synthèse vocale. La synthèse vocale produit une parole artificielle générée par un ordinateur ou une machine à partir d'un texte. L'objectif principal de la synthèse de la parole à partir de textes (TTS pour *Text-To-Speech*) est de produire des sons d'une qualité comparable à la parole humaine. L'expressivité est une caractéristique intrinsèque des phrases prononcées par des humains, et est gérée de manière subconsciente. La synthèse vocale expressive vise à générer, à partir d'une phrase textuelle, une parole qui présente une expressivité correspondant à une émotion, un style, des variations prosodiques, etc.

L'expressivité aide à encoder les sentiments par des expressions non verbales et par le ton de la voix. La théorie veut qu'en général, la capacité d'exprimer des émotions, en particulier la capacité de coder des émotions, peut représenter une composante centrale de la personnalité d'un individu parce que la communication émotionnelle joue un rôle vital dans l'interaction et le développement des relations [Friedman, 1979]. De plus, en tant que style personnel, l'expression des émotions est relativement cohérente dans différentes situations et tout au long du processus de développement [Allport and Vernon, 1933].

Ce travail de thèse étudie le transfert des caractéristiques de l'expressivité dans un contexte de synthèse de parole multilocuteur. L'objectif est de transférer des caractéristiques de l'expressivité estimées à partir de signaux de parole expressive d'un locuteur, sur la voix d'un autre locuteur pour lequel on ne dispose pas de données expressives. Les approches présentées sont conçues pour des systèmes de synthèse de parole multilocuteurs prenant en entrée, en plus du texte, des informations sur le locuteur et sur la classe d'expressivité. Le travail présenté se concentre sur la création de représentations latentes démêlées représentant respectivement le locuteur et l'expressivité. L'un des principaux défis est de transférer les attributs expressifs tout en gardant la voix du locuteur cible inchangée. Le travail se concentre sur la langue française, et considère comme classes d'expressivité des classes d'émotions bien définies.

B.2 Données et prétraitements

Les systèmes de synthèse vocale doivent apprendre la fonction de transformation permettant de produire la forme d'onde vocale à partir d'un texte d'entrée donné. Dans cette section, nous présentons les jeux de données de langue française utilisés ; et nous décrivons les prétraitements appliqués sur les données textuelles et sur les données de parole.

B.2.1 Corpus de synthèse de parole

Cette section présente les corpus de synthèse de parole utilisés.

Corpus Caroline. Le corpus Caroline est un corpus de parole expressive audiovisuelle en langue française qui a été conçu dans l'équipe Multispeech pour fabriquer des systèmes de synthèse de parole audiovisuelle expressive [Dahmani et al., 2019]. Il est constitué de parole neutre et de parole expressive. Six émotions ont été enregistrées : joie, tristesse, peur, surprise, dégoût et colère. Une actrice semi-professionnelle a prononcé 2000 phrases en mode neutre (soit environ 3 heures de parole), et 500 phrases (environ 1 heure de parole) pour chaque émotion. Tous les signaux vocaux ont été enregistrés à 16 kHz.

Corpus Synpaflex. Le corpus Synpaflex [16] a été développé pour l'analyse de l'expressivité. Il correspond à la lecture de livres audio de différents styles littéraires, par une seule locutrice française [Sini et al., 2018]. Les segments de discours ont été annotés manuellement en terme d'émotion. Six émotions de base [Ekman, 1999] ont été considérées (tristesse, colère, peur, joie, surprise et dégoût), plus deux autres pour représenter le contenu de différents livres, à savoir l'ironie et la menace. L'intensité de chaque émotion est indiquée sur une échelle de 1 à 3 ; par exemple, l'intensité légèrement en colère, 1, en colère 2, et fortement en colère 3. Tous les signaux vocaux sont stockés à un taux d'échantillonnage de 44,1 kHz.

Corpus Lisa. Le corpus Lisa a été développé en interne dans l'équipe Multispeech. Il a été initialement construit pour la synthèse de parole neutre par concaténation d'unités et a été particulièrement utilisé dans le système de synthèse de parole Soja. Le corpus est composé de 1812 phrases extraites du journal "Le Monde". Le corpus a été enregistré par une voix féminine dans une pièce calme. Il correspond à environ 3 heures de parole et a été enregistré à 16 kHz.

Corpus Simple4All Tundra. Le corpus Simple4All Tundra [17] est un corpus multilingue conçu pour la recherche sur la synthèse vocale [Stan et al., 2013]. Le corpus comprend environ 60 heures de données vocales recueillies dans des livres audio en 14 langues. Nous avons utilisé le sous-ensemble correspondant à la langue française, enregistré dans une voix neutre. Cela correspond à environ 90 minutes de données vocales.

16. `http://synpaflex.irisa.fr/fr/`
17. `https://simple4all.org/`

Corpus Siwis. Le corpus Siwis a été conçu pour la construction de systèmes de synthèse de parole en français [Honnet et al., 2017]. Il correspond a plus de 10 heures de discours, et a été enregistré par une actrice française, avec des mots accentués dans différents contextes. Des instructions étaient présentées à la locutrice de langue maternelle française, telles que l'accentuation d'un mot spécifique dans une phrase, la lecture du chapitre avec des émotions, et de longues pauses entre les phrases. Nous avons utilisé la partie neutre de ce corpus pour nos expériences (soit 5 heures de parole).

B.2.2 Pré-traitment du texte

Nous avons étudié le transfert d'expressivité dans deux cadres : synthèse de parole paramétrique et synthèse de parole de bout en bout. Dans l'approche paramétrique, la première étape consiste à convertir le texte brut en une représentation qui transmettra les informations linguistiques, phonétiques et prosodiques, résultant en une séquence d'étiquettes contextuelles (également appelées caractéristiques contextuelles) [King, 2011]. Ces étiquettes au format HTS [Zen, 2006] intialement dévelopés pour les outils de synthèse vocale basés sur les HMM (tel que HTS), sont couramment utilisés dans les systèmes de synthèse vocale paramétrique. Nous avons utilisé Soja, un outil de synthèse vocale développé par l'équipe Multispeech du laboratoire LORIA à Nancy, pour le traitement du texte et la génération des étiquettes contextuelles.

Pour l'apprentissage, après avoir généré des étiquettes contextuelles pour le texte, les formes d'onde vocales et les étiquettes contextuelles sont alignées au niveau des phonèmes avec l'outil d'alignement forcé basé sur le toolkit HTK [Young, 1994]. L'alignement fournit les temps début et fin de chaque segment phonétique ; ces informations temporelles sont utilisées pour entraîner le modèle de durée.

Dans le cas de synthèse de parole de bout en bout, le pré-traitement se limite à la conversion graphèmes-phonèmes du texte ; l'alignement temporel n'est pas nécessaire. Là aussi, nous avons utilisé l'outil Soja pour convertir le texte en séquence de phonèmes ; et nous avons attribué des identifiants entiers à chaque phonème dans la plage de 1 à 36 et l'identifiant 37 pour représenter un silence (quelle que soit sa position dans la phrase).

B.2.3 Pré-traitement du signal de parole

Pour le paramétrage du signal acoustique pour les approches paramétriques, nous avons utilisé le vocodeur WORLD [Morise et al., 2016]. Nous avons utilisé ce vocodeur pour calculer 187 caractéristiques acoustiques pour chaque trame temporelle (toutes les 5 ms), à savoir 180 caractéristiques spectrales (60 coefficients MGC (*Mel Generalized Cepstrum*) avec dérivées première et seconde), trois paramètres liés à la fréquence fondamentale (logarithme de la fréquence fondamentale (Lf0), avec dérivées première et seconde) et trois paramètres d'excitation (coefficient d'apériodicité (bap - *band aperiodicity*) et dérivées première et seconde) et une valeur pour l'information voisée-non-voisée (vuv).

Pour les approches de bout en bout, nous avons calculé des spectrogrammes Mel avec 80 bins en utilisant la bibliothèque librosa [18]. Nous avons également utilisé librosa pour

18. https://librosa.org/

le rééchantillonnage de la forme d'onde à la fréquence d'échantillonnage souhaitée [McFee et al., 2015]. Et, nous avons appliqué la STFT avec une taille de FFT de 1024, une taille de fenêtre de 1024, et un décalage de 256 échantillons entre trames.

L'expérimentation initiale avec des systèmes TTS autorégressifs de bout en bout a été menée avec un taux d'échantillonnage de 16 kHz. Pour les approches à gros grain, nous avons utilisé et entraîné le vocodeur neuronal Waveglow [Prenger et al., 2019] à partir de zéro sur tous les corpus de synthèse vocale pour le français, à savoir Caroline, Lisa, Siwis, Synpaflex, et Tundra. Ce vocodeur neuronal a été entraîné avec les hyperparamètres du modèle par défaut et jusqu'à 800 000 étapes d'itération, comme présenté dans l'implémentation originale [19].

Pour les systèmes TTS autorégressifs à grain fin, nous avons d'abord rééchantillonné les ensembles de données vocales à 22050 Hz et utilisé le modèle Waveglow pré-entraîné sur le corpus LJS avec la configuration par défaut, et fourni dans l'implémentation officielle de Waveglow.

Pour les systèmes TTS non-autorégressifs, nous avons choisi comme vocodeur neuronal le modèle Hi-Fi GAN [Kong et al., 2020] pré-entraîné sur le corpus LJS avec la configuration par défaut, et fourni dans l'implémentation officielle [20].

B.3 Evaluation de la synthèse de parole expressive

Jusque dans les années 1990, la pratique la plus courante pour mesurer la qualité de la parole consistait à effectuer des tests d'écoute subjectifs [Rix et al., 2006, Wang et al., 1992]. L'évaluation objective, quant à elle, fournit une estimation approximative de la performance du système TTS. En général, les mesures objective et subjective sont toutes deux utilisées pour évaluer les systèmes de synthèse de parole.

Cette section donne un aperçu des techniques utilisées pour l'évaluation de la qualité du système de synthèse de la parole et de la performance du transfert d'expressivité. Elle décrit successivement les mesures d'evaluation objectives, puis les mesures subjectives résultant de tests d'écoute.

B.3.1 Evaluation objective

Trois mesures sont généralement utilisées pour l'évaluation objective de systèmes de synthèse de parole : MCD (*Mel cepstrum distortion*), Lf0 RMSE (*Root Mean Square Error*), et distorsion BAP (*Band aperiodicity*). Ces mesures visent à estimer un écart (erreur) entre la parole produite et un signal de parole de référence. Les mesures concernent respectivement la distortion spectrale (mesurée à partir des coefficients MGC), l'écart moyen de prédiction du logarithme de la fréquence fondamentale (Lf0), et l'écart sur les coefficients d'apériodicité (bap). Ces caractéristiques acoustiques sont extraites de la forme d'onde de la parole (synthètisée d'une part, et référence d'autre part) à l'aide du vocodeur WORLD.

19. `https://github.com/NVIDIA/waveglow`
20. `https://github.com/jik876/hifi-gan`

Le système de synthèse de la parole peut générer un signal de parole de longueur différente du signal de parole de référence. Dans ce cas, avant de calculer la métrique d'évaluation, nous avons utilisé l'algorithme d'alignement temporel (DTW) [Muller, 2007] pour mettre en correspondance les caractéristiques acoustique du signal de parole généré avec celles du signal de référence [Kearney et al., 2009].

Score de similarité concernant l'expressivité. En ce qui concerne l'expressivité, nous avons entraîné un système de reconnaissance des émotions afin de calculer un score de similarité entre un signal généré et des signaux de référence. Le système de reconnaissance prend en compte les six émotions de la théorie d'Ekman [Ekman, 1999], plus deux autres classes, à savoir l'amusement et la somnolence [Adigwe et al., 2018], et la parole neutre. Nous avons entraîné le système de reconnaissance des émotions sur les corpus de parole décrits dans la section B.2.1, pendant 100 époques avec l'optimiseur Adam [Kingma and Ba, 2015].

Pour l'évaluation, un plongement (*embedding*) de l'expressivité de dimension 256 est extrait à partir de la dernière couche. Le plongement extrait sur la parole générée x_e est comparé (cosinus) à la moyenne $x_{e,mean}$ des plongements des données de référence de la classe d'émotion souhaitée.

Score de similarité concernant le locuteur. La même architecture a été utilisée pour développer un système de reconnaissance du locuteur, qui a lui aussi été entrainé sur les corpus de parole mentionnés dans la section B.2.1, pendant 100 époques avec l'optimiseur Adam. Au total, il y avait 145 locuteurs pour l'apprentissage du modèle, et les données de parole incluaient de la parole expressive pour certains locuteurs.

Pour l'évaluation, un plongement de locuteur de dimension 256 est extrait à partir de la dernière couche. Le plongement extrait sur la parole générée x_s est comparé (cosinus) à la moyenne $x_{s,mean}$ des plongements des données de référence du locuteur souhaité.

B.3.2 Evaluation subjective

L'évaluation subjective est une étape critique dans le développement de systèmes de synthèse de la parole. Elle repose sur l'écoute par des auditeurs de signaux de parole générés par le ou les systèmes de synthèse de parole à évaluer pour des phrases extraites d'un ensemble de test ; les auditeurs notant typiquement la qualité perçue.

L'objectif principal de la thèse est de transférer l'émotion en tant qu'attribut expressif sur la voix du locuteur cible sans altérer les caractéristiques de la voix de ce locuteur. En conséquence, cette section introduit deux tests d'écoute spécifiques pour évaluer la qualité de l'expressivité générée et la qualité perçue de la voix du locuteur cible.

Mean opinion score. Le score MOS (*Mean Opinion Score*) est couramment utilisé dans les tests d'écoute pour évaluer le caractère naturel, l'intelligibilité, ou d'autres facteurs qualitatifs de la parole synthétisée [Wang et al., 2017b, Theis et al., 2016]. Le score MOS est associé à échelle de Likert [Joshi et al., 2015] à 5 points. Pour cette thèse le score MOS fait référence à l'évaluation de la qualité globale d'un stimuli perçue par les auditeurs (en

terme de qualité et de naturel de la phrase écoutée, sans tenir compte de l'expressivité) : de 1 (mauvaise) à 5 (excellente).

MOS locuteur (*speaker similarity MOS*). Un des tests d'écoute consistait à évaluer par un score MOS la similarité de locuteur perçue entre deux stimuli. Chaque énoncé généré par la synthèse de la parole était joué avec un énoncé de référence prononcé par le locuteur cible. Chaque auditeur devait donner un score de 1 (mauvais) et 5 (excellent), en fonction de la similarité perçue entre les locuteurs des deux énoncés. Une instruction précisait de ne pas juger le contenu, la grammaire ou la qualité audio des phrases ; mais au contraire, de se concentrer sur la similarité des locuteurs entre eux.

MOS expressivité (*expressivity similarity MOS*). Il s'agit ici d'évaluer la similirité d'expressivité (emotion) entre deux stimuli. Dans ces tests d'écoute, chaque énoncé généré par la synthèse de la parole été joué avec un énoncé de référence correspondant à l'expressivité cible. Chaque auditeur devait donner un score de 1 (mauvais) et 5 (excellent), en fonction de la similarité perçue entre l'expressivité des deux énoncés. De même, une instruction précisait de ne pas juger le contenu, la grammaire ou la qualité audio des phrases ; mais au contraire, de se concentrer sur la similarité de l'expressivité des deux énoncés.

B.4 Synthèse de parole expressive par approche paramétrique

Les systèmes neuronaux de synthèse de parole par approche paramétrique utilisent des réseaux de neurones pour prédire les caractéristiques acoustiques directement à partir d'un ensemble de caractéristiques textuelles ; la forme d'onde est alors construite, grâce à un vocodeur, à partir de ces caractéristiques acoustiques prédites [Zen et al., 2013, Wu et al., 2016]. Les systèmes comprennent généralement deux réseaux de neurones : un pour prédire la durée des sons, et un autre pour prédire les paramètres acoustiques. De nombreuses approches ont été proposées tant pour générer de la parole neutre [Zen et al., 2013, Lu et al., 2013, Qian et al., 2014, Wu et al., 2015, Hashimoto et al., 2015] que de la parole expressive [Yamagishi et al., 2004, Xue et al., 2018, Li et al., 2018].

Dans le cadre de la synthèse de parole paramétrique, peu d'études pour transférer l'expressivité sur la voix d'un nouveau locuteur ont été publiées [Parker et al., 2018, Chen and Braunschweiler, 2013, Chen et al., 2015, Ohtani et al., 2015]. Le transfert d'expressivité est accompli dans [Ohtani et al., 2015] en créant un espace de "voix propres" (eigenvoices) de la parole neutre ainsi que les contrastes entre parole neutre et expressive. Dans [Chen and Braunschweiler, 2013, Chen et al., 2015] deux sous-espaces adaptatifs (CAT : *Cluster Adaptive Training*) sont produits, l'un pour les locuteurs et l'autre pour l'expressivité.

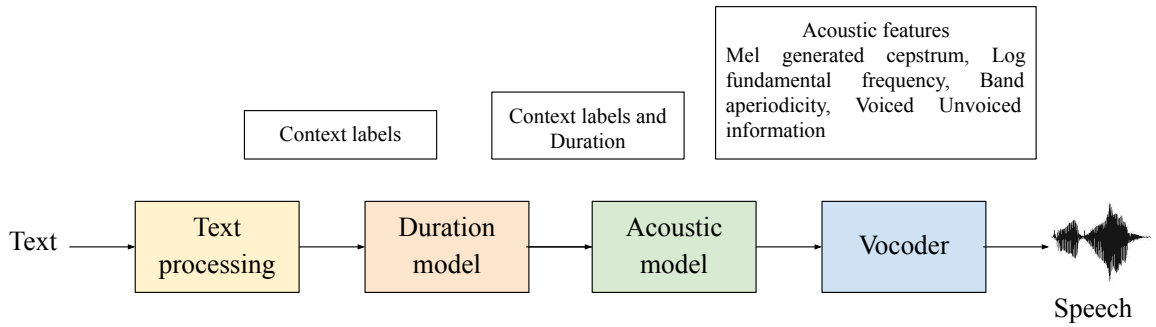

FIGURE B.1 – Cadre de synthèse de parole paramétrique monolocuteur.

B.4.1 Approche proposée

La figure B.1 illustre le cadre général d'un système de synthèse de parole paramétrique monolocuteur. Le premier module traite le texte, le converti en phonèmes, et crée une représentation des informations associées à chaque phonéme (*Context labels*). Le second module prédit la durée de chaque phonème. Le troisième module prédit alors les caractéristiques acoustiques de la parole, comme la fréquence fondamentale, les paramètres spectraux, la durée, l'apériodicité, etc. Ces paramètres sont fournis au vocodeur pour synthétiser la forme d'onde vocale.

Nous proposons trois variantes pour l'implémentation du modèle acoustique, à savoir un autoencodeur variationnel conditionné, un autoencodeur variationnel conditionné avec prise en compte d'un critère de *metric learning* lors de l'apprentissage, et une approche par *inverse autoregressive Flow* (également avec prise en compte d'un critère de *metric learning* lors de l'apprentissage) [Kulkarni et al., 2020a, Kulkarni et al., 2020b]. Les réseaux encodeur et décodeur sont communs aux différentes architectures. L'encodeur est un réseau récurrent qui permet le traitement de séquences d'entrée de longueur variable et crée une représentation latente de dimension fixe. Le décodeur est également un réseau récurrent qui prédit les caractéristiques acoustiques. Dans ce travail, pour prédire la durée (et donc le nombre de trames acoustiques) de chaque phonème, nous avons utilisé un réseau neuronal basé sur un BLSTM. Les architectures mises en oeuvre créent au niveau intermédiaire une représentation latente de l'expressivité. Le transfert d'expressivité peut alors être effectué en interpolant des variables latentes de l'expressivité souhaitée.

Pour conditionner le décodeur, nous avons utilisé, en plus des caractéristiques textuelles, des plongements de locuteurs créés à partir d'un modèle de reconnaissance du locuteur entrainé sur les corpus de synthèse vocale du français (voir la section B.2.1).

Modèle : autoencodeur variationnel (RCVAE)

Pour l'architecture RCVAE (*Recurrent Conditionnal Variationnal Auto-Encoder*), nous avons utilisé un réseau encodeur basé sur un BLSTM, qui prend en entrée une séquence de caractéristiques acoustiques, x. Les sorties de la couche BLSTM sont transmises à une couche entièrement connectée pour calculer le vecteur moyenne (μ) et la variance (σ^2).

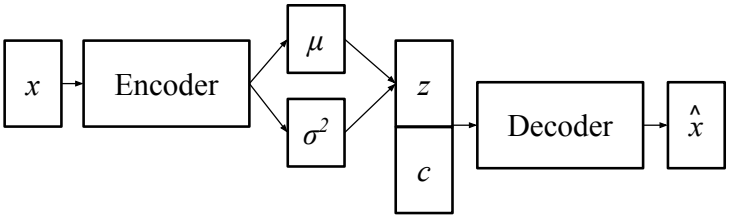

FIGURE B.2 – Architecture RCVAE utilisée pour l'apprentissage du modèle acoustique. Ici, x est une séquence de caractéristiques acoustiques à reconstruire sous la forme $\hat{x}$, c est une condition (caractéristiques textuelles, plongement du locuteur), μ et σ^2 sont des paramètres de moyenne et de variance fournis par l'encodeur, et utilisés pour générer la variable latente z.

Ces valeurs sont ensuite utilisées pour décrire la variable latente de l'encodeur, z.

Le décodeur est également constitué de couches BLSTM. L'entrée du décodeur est la variable latente z et la condition c. Ici, la condition c correspond aux caractéristiques textuelles, aux informations de durée et au plongement du locuteur. Le décodeur génère la séquence de caractéristiques acoustiques $\hat{x}$, comme le montre la figure B.2. La fonction de coût lors de l'apprentissage correspond au coût de reconstruction plus un terme de régularisation défini avec la divergence de Kullback-Leibler (KL).

Dans la phase d'inférence, on échantillonne z dans l'espace latent et on le fourni au décodeur avec la condition c (caractéristiques textuelles et plongement du locuteur), afin d'obtenir les caractéristiques acoustiques pour la synthèse vocale.

Modèle : autoencodeur variationnel et *metric learning* (RVCAE+N-pair)

Pour cette approche, nous gardons l'autoencodeur variationnel décrit ci-dessus, en ajoutant, lors de l'apprentissage, un second terme de régularisation basé sur le *metric learning*, en complément de la divergence de Kullback-Leibler mentionnée précédemment. Ici, nous utilisons le critère *multi-class N-pair loss* [Sohn, 2016] appliqué sur les classes d'émotions. Le principe est de favoriser le rapprochement des plongements d'une même émotion tout en les eloignant des plongements des autres émotions. Cela aide à mieux séparer les clusters correspondant à chaque émotion dans l'espace latent, et améliore les performances de transfert d'expressivité.

Modèle : *inverse autoregressive Flow* et *metric learning* (IAF+N-Pair)

Dans cette approche, représentée sur la figure B.3 nous effectuons une inférence variationnelle basée sur une succession de transformations normalisées (*Flow*) qui vient remplacer la divergence de Kullback-Leibler utilisée dans les deux approches précédentes. Comme précédemment, l'encodeur et le décodeur reposent sur des réseaux BLSTM. La sortie de l'encodeur est utilisée pour calculer la moyenne initiale μ_0, la variance initiale σ_0^2 et la sortie cachée h. Ensuite, z_0 ainsi que la sortie cachée h sont soumis au processus IAF pour obtenir z_K après K transformations. Ainsi, lors de l'apprentissage la régularisation par divergence KL des approches précédentes est remplacée par la prise en compte de la

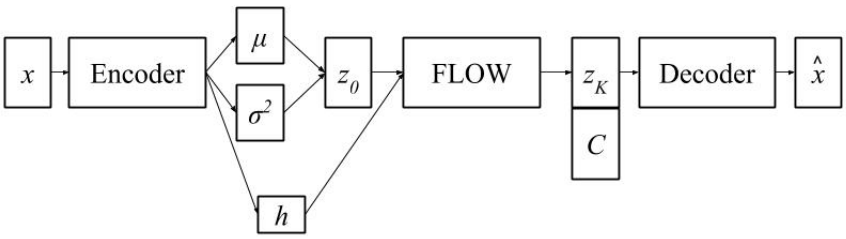

FIGURE B.3 – Architecture *inverse autoregressive Flow*. x est une séquence de caractéristiques acoustiques à reconstruire sous la forme $\hat{x}$, c est une condition (caractéristiques textuelles, plongement du locuteur), μ et σ^2 sont des paramètres de moyenne et de variance fournis par l'encodeur, et utilisés pour générer la variable latente z_0. Le processus IAF produit alors z_K après application de K transformations.

modification de la densité de probabilité $logQ(z_K|x)$ due aux K transformations. De plus, lors de l'apprentissage, nous utilisons également le critère *multi-class N-pair loss* [Sohn, 2016] appliqué sur les classes d'émotions.

Dans la phase d'inférence, on échantillonne z_0 dans l'espace latent pour obtenir z_k après transformation de Flow. Ensuite, z_k est donné au décodeur avec la condition c (caractéristiques textuelles et plongement du locuteur), afin d'obtenir les caractéristiques acoustiques pour la synthèse vocale.

Spécification de l'expressivité

L'expressivité ciblée e est spécifiée en prenant la valeur moyenne de la variable latente (plongements) pour la classe d'expressivité concernée ($z_{e,mean}$ ou $z_{0,e,mean}$ selon l'approche).

Le plongement du locuteur cible est spécifié dans la condition c en entrée du décodeur, en complément des informations textuelles, et des informations de durée prédites par des modèles de durées spécifiques à chaque classe d'expressivité.

B.4.2 Résultats

Le tableau B.1 présente les résultats de l'évaluation des systèmes de synthèse de parole paramétriques (sans transfert d'expressivité) ; et le tableau B.2 présente les résultats des mêmes systèmes utilisés avec transfert d'expressivité. Suite à un test d'écoute informel, le modèle IAF sans prise en compte du critère *multiclass N-pair loss* lors de l'apprentissage n'a pas été inclus dans les évaluations subjectives. Les scores MOS, MOS locuteur, et MOS expressivité sont affichés avec les intervalles de confiance à 95 % associés.

Les résultats du tableau B.1 montrent que le modèle IAF appris avec le critère *multi-class N-pair loss* est plus performant que les modèles à base d'encodeur autovariationnel qu'ils soient appris avec ou sans ce critère. Les mesures objectives (MCD, BAP et Lf0 RMSE) sont alignés avec les scores MOS de l'évaluation subjective. Les scores de similarité

Tableau B.1 – Evaluation de la performance des systèmes de synthèse de parole paramétrique.

Modèle	MOS	MCD	BAP	Lf0 RMSE	Similarité locuteur	Similarité expressivité
RCVAE	2.62 ± 0.5	5.45	1.54	19.25	0.79	0.87
RCVAE+N-pair	2.97 ± 0.4	5.34	1.25	18.91	0.81	0.90
IAF+N-pair	**3.02 ± 0.4**	**5.21**	**1.24**	**17.84**	**0.81**	**0.91**

Tableau B.2 – Evaluation de la performance des systèmes de synthèse de parole paramétrique, lors du transfert d'expressivité.

Modèle	MOS locuteur	MOS expressivité	Similarité locuteur	Similarité expressivité
RCVAE	2.40 ± 0.2	1.53 ± 0.4	0.75	0.24
RCVAE+N-pair	2.86 ± 0.2	1.93 ± 0.3	0.76	0.26
IAF+N-pair	**2.93 ± 0.3**	**2.03 ± 0.3**	**0.77**	**0.27**

montrent également que ces modèles ont conservé la voix du locuteur cible et l'expressivité cible.

Les résultats du tableau B.2, pour le transfert d'expressivité, montrent que les scores de similarité du locuteur restent proches de ceux des systèmes sans transfert (cf. tableau B.1), contrairement au score de similarité de l'expressivité.

Concernant l'autoencodeur variationnel (RCVAE), la prise en compte du critère *multiclass N-pair loss* lors de l'apprentissage améliore notablement les performances en particulier lors du transfert d'expressivité. Les résultats subjectifs du tableau B.2 montrent également que l'approche *inverse autoregressive Flow* avec le critère *multiclass N-pair loss*, conduit aux meilleurs performances d'expressivité perçue dans la voix du locuteur souhaité. Les mesures d'évaluation objectives proposées, basées sur la similarité cosinus, sont en accord avec les mesures d'évaluation subjectives.

B.4.3 Conclusion

Nous avons développé des systèmes de synthèse de parole paramétriques, et montré que l'ajout d'un critère de *metric learning* lors de l'apprentissage améliore la représentation de l'expressivité dans l'espace latent ; qui se traduit par une meilleure séparation des classes d'expressivité dans l'espace latent.

Alors que les autoencodeurs variationnels reposent sur une approximation par une distribution gaussienne de moyenne nulle et de variance unitaire dans l'espace latent, l'approche *inverse autoregressive Flow* contruit des distributions postérieures en utilisant une cascade de transformations [Nalisnick et al., 2016, Salimans et al., 2015, Tran et al., 2015]. Nous avons ainsi proposé et implémenté un modèle acoustique reposant sur l'approche *inverse autoregressive Flow* et le critère *multiclass N-pair loss* lors de l'ap-

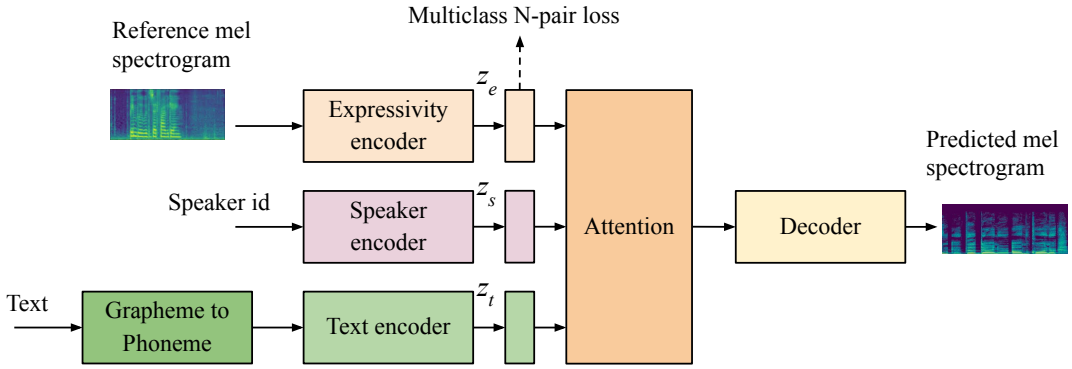

FIGURE B.4 – Système de synthèse de parole expressive multilocuteur de bout en bout basé sur le transfert d'expressivité à gros grain.

prentissage. Bien que l'approche de synthèse paramétrique semble ne permettre qu'un faible transfert de l'expressivité, les résultats montrent que le dernier modèle proposé (à base de *inverse autoregressive Flow*) donne les meilleures performances, en surpassant les approches à base d'autoencodeurs variationnels.

B.5 Synthèse de parole de bout en bout autorégressive

Comme dans beaucoup d'autres domaines, des approches de bout en bout ont été proposées pour la synthèse de la parole (e.g., [Shen et al., 2018, Wang et al., 2017b, Arik et al., 2017, Ren et al., 2019, Sotelo et al., 2017]). Parmi celles-ci, l'architecture Tacotron repose sur l'apprentissage de séquence à séquence avec mécanisme d'attention [Wang et al., 2017b].

Dans cette section, nous présentons des systèmes de synthèse de parole de bout en bout basés sur le système Tacotron 2 [Shen et al., 2018]. Le système Tacotron 2 consiste en un réseau récurrent de séquence à séquence qui prédit des spectrogrammes Mel à partir des caractéristiques textuelles. Nous avons étendu le modèle Tacotron 2 avec un module d'attention, un encodeur d'expressivité et un encodeur de locuteur, afin d'implémenter un système de synthèse de parole expressif multilocuteur de bout en bout.

B.5.1 Transfert d'expressivité à gros grain

L'objectif principal de l'approche à gros grains est de fournir un plongement d'expressivité de dimension fixe, global à l'énoncé. Le système proposé est représenté sur la figure B.4. Le texte est d'abord converti en une suite de phonèmes. L'encodeur de texte est composé de couches convolutionnelles et d'une couche BLSTM, et crée un plongement z_t à partir de la suite de phonèmes. L'identité du locuteur est fournie sous forme d'index à l'encodeur locuteur qui en extrait le plongement z_s de dimension fixe. L'expressivité cible est spécifiée par un exemple de parole dont le spectrogramme Mel est traité par l'encodeur d'expressivité pour fournir le plongement z_e. Deux approches ont été utilisées pour

encoder l'expressivité : le *Global Style Token* (GST) [Wang et al., 2018] et le *Variational Autoencoder* (VAE) [Zhang et al., 2019].

Comme illustré sur la figure B.4, z_t, z_e et z_s sont fournis au module d'attention. Cela aide le système de synthèse de parole de bout en bout à déterminer l'alignement entre la séquence de phonèmes et le spectrogramme Mel souhaité. Les sorties des encodeurs et le vecteur d'attention sont traités par le décodeur pour prédire le spectrogramme Mel trame par trame.

Le système est entraîné avec le critère *multiclass N-pair loss* appliqué sur les sorties de l'encodeur d'expressivité, en plus des coûts de reconstruction et de la divergence Kullback-Leibler pour l'inférence variationnelle.

Pour le transfert de l'expressivité, nous utilisons les moyennes des variables latentes pré-calculées pour chaque classe d'expressivité. C'est pour cela que nous avons proposé d'inclure le critère *multiclass N-pair loss* lors de l'apprentissage afin d'obtenir des clusters d'expressivité bien séparés dans l'espace latent.

B.5.2 Transfert d'expressivité à grain fin

Cette approche de prise en compte de l'expressivité à grain fin permet d'apprendre une représentation de l'expressivité en fonction du phonème ou d'une partie de l'énoncé. Deux variantes d'architecture sont proposées, représentées sur les figures B.5 et B.6.

Trois stratégies d'attention sont mises en oeuvre dans ces deux modèles. Les sorties de l'encodeur d'expressivité sont traitées par une couche *self-attention*, suivie d'un autoencodeur GMVAE (*Gaussian mixture variational autoencoder*) [Dilokthanakul et al., 2016] pour extraire une représentation globale z_e de l'expressivité. Un module *positional encoder* génère des poids d'attention expressive à partir des sorties de l'encodeur de texte et de l'encodeur de l'expressivité de référence, et donc des informations d'expressivité locales (associées aux phonèmes). Enfin, le décodeur est précédé d'une couche *location sensitive attention* qui aligne les données d'entrée (phonèmes) et le spectrogramme Mel. Ces couches d'attention à différents niveaux aident à la découverte des caractéristiques d'expressivité locales et globales. Le reste de l'architecture du système de synthèse multilocuteur expressif à grain fin est le même que celui expliqué dans la section précédente (approche à gros grain).

Le premier modèle à grain fin proposé (modèle I) est illustré sur la figure B.5. Le plongement locuteur Z_s, le plongement expressif global z_e, et les sorties du module *positional attention* $z_{e,attention}$ sont concaténées et fournies au module *location sensitive attention*.

Le second modèle à grain fin proposé (modèle II) est illustré sur la figure B.6. Dans ce modèle, le plongement locuteur Z_s, le plongement expressif global z_e, les sorties du module *positional attention* $z_{e,attention}$ et le plongement du texte z_t sont concaténées et fournies au module *location sensitive attention*.

B.5.3 Resultats

L'évaluation subjective (score MOS) a été réalisée avec 12 participants de langue maternelle française. Chaque auditeur a évalué 18 stimuli pour chaque système de synthèse de parole de bout en bout autorégressif.

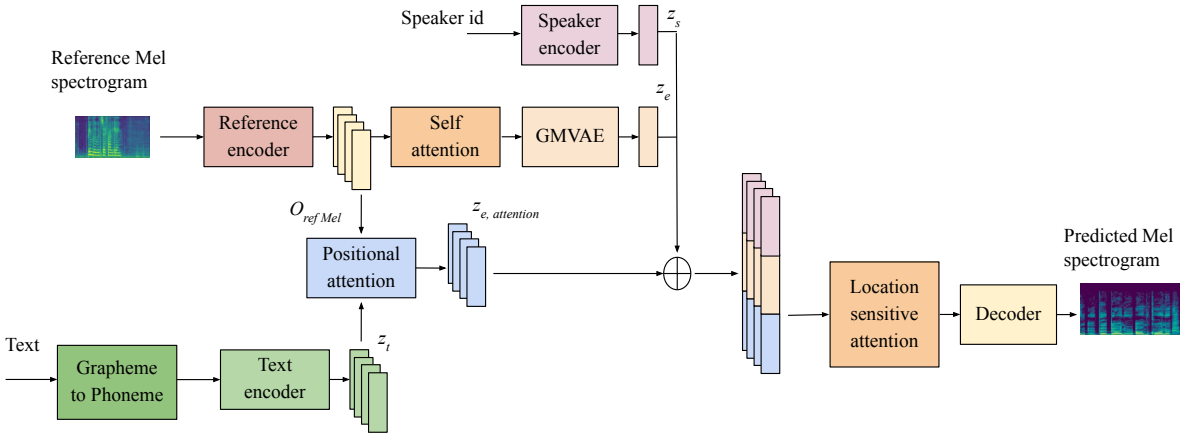

FIGURE B.5 – Système de synthèse de parole expressive multilocuteur de bout en bout basé sur le transfert d'expressivité à grain fin - modèle I.

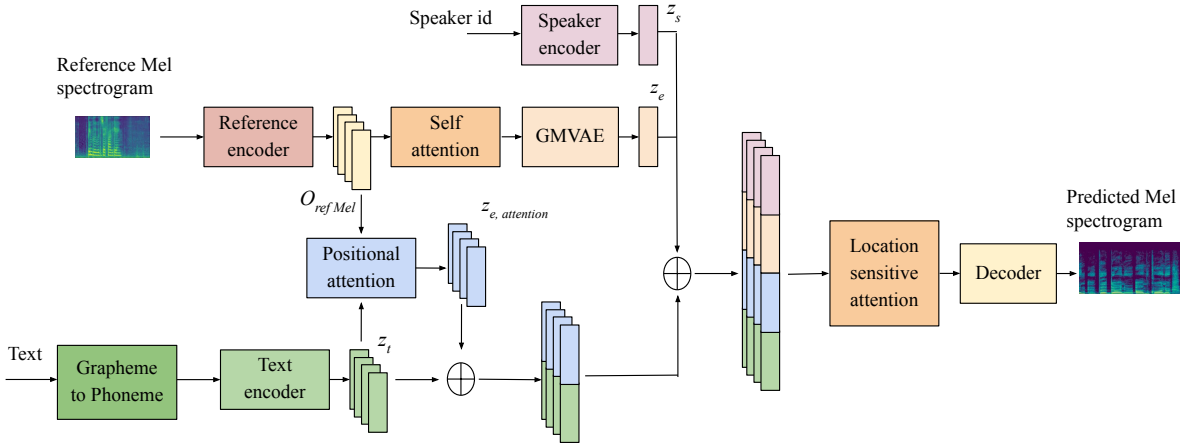

FIGURE B.6 – Système de synthèse de parole expressive multilocuteur de bout en bout basé sur le transfert d'expressivité à grain fin - modèle II.

Le tableau B.3 détaille les performances des systèmes de synthèse de parole autorégressifs proposés. Pour l'approche à gros grain, et pour les deux codeurs d'expressivité (GST VAE), la prise en compte du critère *multiclass N-pair loss* lors de l'apprentissage améliore les performances ; les meilleures performances (score MOS) sont obtenues avec l'encodeur GST pour l'expressivité. Les mesures objectives sont consistentes avec les résultats de l'évaluation subjective (MOS).

Pour l'approche à grain fin, les mesures objectives MCD et BAP sont moins bonnes que celles obtenues avec l'approche à gros grain ; par contre les mesures Lf0 RMSE sont bien meilleures. L'évaluation subjective (score MOS) montre que c'est l'approche à grain fin qui conduit à la meilleure qualité de parole générée, le modèle II étant très légèrement meilleur.

Pour l'évaluation MOS locuteur et MOS expressivité lors du transfert d'expressivité,

Tableau B.3 – Evaluation de la performance des systèmes de synthèse de parole de bout en bout autorégressifs.

Modèle	MOS	MCD	BAP	Lf0 RMSE	Similarité locuteur	Similarité expressivité
E2E GST	3.51 ± 0.3	4.36	0.83	18.23	0.91	0.90
E2E VAE	3.38 ± 0.4	4.76	0.88	18.79	0.92	0.91
E2E GST N-pair	3.72 ± 0.4	4.33	0.73	18.18	0.92	0.90
E2E VAE N-pair	3.47 ± 0.3	**4.21**	**0.72**	18.57	**0.93**	0.93
FG model I	3.81 ± 0.2	4.49	0.82	16.68	0.81	0.93
FG model II	**3.85** ± 0.2	4.50	0.83	**16.66**	0.83	**0.94**

Tableau B.4 – Evaluation de la performance des systèmes de synthèse de parole de bout en bout autorégressifs, lors du transfert d'expressivité en mode parallèle.

Modèle	MOS locuteur	MOS expressivité	Similarité locuteur	Similarité expressivité
E2E GST	2.57 ± 0.2	3.05 ± 0.2	0.62	0.53
E2E VAE	2.71 ± 0.3	3.12 ± 0.2	0.69	0.55
E2E GST N-pair	2.65 ± 0.2	3.15 ± 0.4	0.65	0.55
E2E VAE N-pair	**2.90 ± 0.3**	3.33 ± 0.4	**0.71**	0.57
FG model I	2.75 ± 0.3	3.40 ± 0.3	0.68	0.59
FG model II	2.83 ± 0.3	**3.58** ± 0.2	0.68	**0.61**

chaque auditeur a noté cinq stimuli pour chaque paire locuteur-émotion. Le tableau B.4 présente les performances du transfert d'expressivité dans un cadre de transfert parallèle. Là aussi, pour l'approche à gros grain et pour les deux codeurs d'expressivité (GST & VAE), la prise en compte du critère *multiclass N-pair loss* lors de l'apprentissage améliore les performances ; les meilleures performances (scores MOS) sont obtenues avec l'encodeur VAE pour l'expressivité.

Parmi tous les systèmes autorégressifs, c'est le système autorégressif à gros grain avec l'encodeur VAE qui conduit aux meilleurs scores de locuteur (MOS locuteur et similarité locuteur). En ce qui concerne les scores d'expressivité, les modèles à grain fin sont ceux qui ont obtenu les meilleures performances (MOS expressivité et similarité expressivité). Là aussi les meilleures performances sont obtenues par le modèle II, et montrent l'intérêt d'utiliser explicitement les plongements du texte z_t en complément des poids du mécanisme d'attention $z_{e,attention}$, et des plongements globaux de l'expressivité z_e et du locuteur z_s.

Des expériences ont également été menées dans un cadre de transfert non parallèle. Les résultats obtenus, non détaillés dans ce résumé, ont conduit à de moins bonnes performances en ce qui concerne le transfert d'expressivité (MOS expressivité et similarité

expressivité).

B.5.4 Conclusion

Dans cette section, nous avons proposé différents systèmes de synthèse de parole de bout en bout pour transférer l'expressivité dans un contexte multilocuteur. Nous avons étudié deux approches de représentation de l'expressivité : une à gros grain, et une à grain fin ; et nous avons détaillé les résultats d'évaluation pour le transfert d'expressivité dans un cadre de transfert parallèle, c'est-à-dire un même contenu textuel pour le texte à synthétiser (fourni en entrée de l'encodeur de texte), et le signal expressif de référence (fourni en entrée de l'encodeur d'expressivité).

Dans les approches à gros grain, l'utilisation du critère *multiclass N-pair loss* lors de l'apprentissage conduit à des modèles plus performants. En ce qui concerne le transfert de l'expressivité, l'approche qui exploite l'encodeur d'expressivité basé sur VAE généralise mieux la représentation des émotions que l'approche GST.

Nous avons également proposé des architectures à grain fin, dans lesquelles les informations provenant de l'encodeur d'expressivité et de l'encodeur de texte sont utilisées pour construire des poids d'attention expressifs. Ces poids d'attention portent alors des informations différentes pour les différents phonèmes. De plus, la concaténation des plongements de texte avec les poids d'attention de l'expressivité semble jouer un rôle important dans la qualité du signal de synthèse généré et dans le transfert de l'expressivité.

B.6 Synthèse de parole de bout en bout non-autorégressive

Les systèmes de synthèse de parole non-autorégressifs ont abouti à des performances similaires à celles des systèmes autorégressifs, comme par exemple Glow-TTS [Kim et al., 2020], Grad-TTS [Popov et al., 2021], Diff-TTS [Jeong et al., 2021]. Le succès récent des modèles probabilistes de diffusion en tant que modèle génératifs profonds a ouvert la voie à leur utilisation pour générer des spectrogrammes Mel. Les systèmes Grad-TTS et Diff-TTS sont basés sur des modèles probabilistes de diffusion et une prédiction explicite des durées. Peu de recherches ont été menées sur l'expressivité dans les systèmes de synthèse de parole non-autorégressifs [Lee et al., 2021, Lee, 2021, Shah et al., 2021]. Dans cette section, nous présentons l'ajout d'un encodeur d'expressivité aux systèmes de synthèse de parole non-autorégressifs reposant sur Glow (*Generative Flow*) [Kim et al., 2020] et sur des modèles probabilistes de diffusion (DPM) [Popov et al., 2021].

B.6.1 Approches proposées pour transfert d'expressivité

Les approches proposées reposent sur les systèmes Grad-TTS et Glow-TTS pour lesquels nous modifions l'encodeur de texte, et nous ajoutons un encodeur de locuteur et un encodeur d'expressivité pour en faire des systèmes de synthèse de parole expressive multilocuteurs. Le plongement d'expressivité z_e et le plongement de locuteur z_s sont concaténés à chacun des éléments de plongement du texte. La recherche d'un alignement monotone

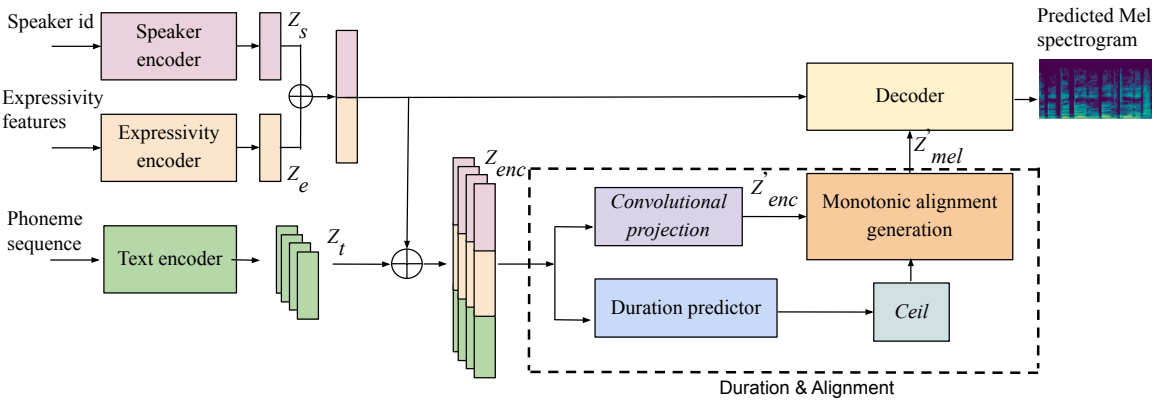

FIGURE B.7 – Architecture non-autorégressive pour la synthèse de parole expressive en conditionnant explicitement le décodeur et le prédicteur de durée avec les plongements d'expressivité et du locuteur.

entre la séquence de phonèmes et le spectrogramme Mel repose sur la méthode de Viterbi [Rabiner, 1989] et est également conditionnée par les plongements de locuteur et d'expressivité (figure B.7).Les plongements d'expressivité et de locuteur (z_e et z_s) sont également fournis comme entrées de canaux supplémentaires au décodeur (Glow ou DPM selon les modèles).

L'identité du locuteur est fournie sous forme d'un entier à l'encodeur locuteur, qui utilise la fonction nn.Embedding() de la librairie PyTorch [21] pour déterminer le plongement locuteur z_s de dimension 80.

En ce qui concerne le plongement d'expressivité, trois approches différentes ont été mises en oeuvre. La première technique (désignée "one-hot" dans la suite) repose sur l'identification de chaque émotion par un entier, et le calcul d'un plongement avec la même approche que celle précédemment décrite pour le calcul du plongement locuteur.

La deuxième approche consiste à extraire un plongement d'expressivité à partir d'un spectrogramme Mel de référence (d'un signal de parole correspondant à l'expressivité désirée). Le modèle repose sur un CRNN (*convolutional recurrent neural network*) et un méchanisme d'attention (SASP : *self-attentive statistical pooling*). Ce plongement est désigné dans la suite *attention-pool*.

La troisième approche exploite un ensemble de caractéristiques traditionnellement utilisées dans les systèmes de reconnaissance des émotions [Arias-Vergara et al., 2017, Vásquez-Correa et al., 2018, Orozco-Arroyave et al., 2018, Dehak et al., 2007], qui sont fournies en entrée d'un DNN [Peng et al., 2021]. Les caractéristiques utilisées correspondent à différentes échelles de temps (voir détails en Annexe A de la thèse), d'où la dénomination *multiscale* pour ce plongement.

Tableau B.5 – Evaluation de la performance des systèmes de synthèse de parole de bout en bout non-autorégressifs.

Modèle	MOS	MCD	BAP	Lf0 RMSE	Similarité locuteur	Similarité expressivité
Glow base	—	4.59	1.00	18.40	0.79	0.94
Glow-one-hot-embedding	3.63 ± 0.1	4.57	**1.11**	18.80	0.79	0.81
Glow-attention-pool	3.72 ± 0.2	4.56	1.13	18.65	**0.81**	0.79
Glow-multiscale	3.86 ± 0.1	4.53	1.15	**18.24**	0.80	0.76
DPM-one-hot-embedding	3.68 ± 0.1	4.61	1.20	18.94	0.77	0.82
DPM-attention-pool	3.85 ± 0.2	4.54	1.24	18.85	0.79	**0.83**
DPM-multiscale	**3.94 ± 0.2**	**4.51**	1.27	18.88	0.71	0.78

Tableau B.6 – Evaluation de la performance des systèmes de synthèse de parole de bout en bout non-autorégressifs, lors du transfert d'expressivité en mode parallèle.

Modèle	MOS locuteur	MOS expressivité	Similarité locuteur	Similarité expressivité
Glow base	—	—	0.73	0.31
Glow-one-hot-embedding	3.12 ± 0.1	2.81 ± 0.2	0.68	0.35
Glow-attention-pool	3.20 ± 0.1	2.73 ± 0.1	0.70	0.31
Glow-multiscale	**3.34 ± 0.2**	**2.87 ± 0.1**	**0.73**	0.30
DPM-one-hot-embedding	3.13 ± 0.3	2.78 ± 0.2	0.68	0.35
DPM-attention-pool	3.06 ± 0.2	2.73 ± 0.1	0.62	0.32
DPM-multiscale	3.15 ± 0.1	2.81 ± 0.2	0.69	0.34

B.6.2 Resultats

20 participants de langue maternelle française ont participé à l'évaluation subjective des systèmes de synthèse de parole non-autorégressifs. Chaque participant a écouté six stimuli vocaux pour chacun des systèmes de synthèse non-autorégressifs à évaluer, extraits d'un ensemble global de 176 stimuli vocaux avec un contenu textuel unique.

Le tableau B.5 présente les performances de synthèse vocale avec les modèles non-autorégressifs. Pour les deux décodeurs (Glow & DPM), l'utilisation de caractéristiques de l'émotion extraites par "*attention-pooling*" d'une référence acoustique conduit à de meilleurs scores MOS que l'indication directe de l'émotion (*one-hot*). L'utilisation d'un encodeur d'expressivité multi-échelle (*multiscale*) conduit aux meilleures performances pour les deux décodeurs (Glow & DPM).

Pour évaluer les performances dans un contexte de transfert d'expressivité, une évaluation subjective a été menée auprès de 20 participants de langue maternelle française.

21. https ://pytorch.org/

Les résultats du tableau B.6 montrent que dans le cadre d'un transfert d'expressivité en mode parallèle, c'est aussi l'encodeur d'expressivité multi-échelles (*multiscale*) qui est le plus performant. Les résultats montrent également que l'approche Glow-multiscale conduit à des performances légèrement meilleures tant pour le MOS expressivité que pour le MOS locuteur, que l'approche DPM-multiscale en ce qui concerne le transfert d'expressivité.

Des expériences ont également été menées dans un cadre de transfert non parallèle. Les résultats, non détaillés dans ce résumé, on montré une similarité locuteur un peu plus élevée que celle observée dans le cadre d'un transfert en mode parallèle, par contre la similarité expressivité était un peu moins bonne.

B.6.3 Conclusion

Nous avons exploré l'utilisation de systèmes de synthèse de parole non-autorégressifs pour le transfert d'expressivité avec deux types de décodeurs, à savoir Glow et DPM. En outre, nous avons étudié différents types de spécification de l'expressivité cible et différentes manières d'encoder l'expressivité. Les résultats montrent que l'encodeur d'expressivité multi-échelle est l'approche la plus performante pour encoder l'expressivité ciblée.

B.7 Conclusion

Comme première étape pour traiter le transfert d'expressivité, nous avons proposé d'implémenter une approche paramétrique de synthèse de parole expressive multilocuteur. Cela a amené à ajouter au système un encodeur de locuteur et un encodeur d'expressivité qui délivrent des plongements de locuteur et d'expressivité qui sont alors fournis au décodeur pour la génération des caractéristiques acoustiques. Dans ce cadre, nous avons utilisé des autoencodeurs variationnels, ainsi qu'une architecture *inverse autoregressive Flow*.

Nous avons également proposé d'ajouter un critère de *metric learning*, en l'occurrence le critère *multiclass N-pair loss*, lors de l'apprentissage du systèmes. L'introduction de ce critère permet de mieux séparer les espaces latents correspondant à chacune des émotions, ce qui conduit à de meilleures performances de synthèse de parole expressive. Les meilleures performances ont été obtenues avec le système *inverse autoregressive Flow* appris avec le critère *multiclass N-pair loss*.

Nous avons ensuite étudié plusieurs systèmes de synthèse de parole de bout en bout, en commençant par des approches autorégressives basées sur l'architecture Tacotron 2. Deux variantes d'encodage de l'expressivité à gros grain (i.e., global à l'énoncé) ont été explorées : une reposant sur le *Global Style Token* (GST) et l'autre sur un autoencodeur variationnel (VAE). Dans les deux cas, l'utilisation du critère *multiclass N-pair loss* s'est avérée bénéfique. Nous avons également étudié une approche à grain fin pour une modélisation locale des caractèristiques de l'expressivité (au lieu d'un unique plongement global à l'énoncé) grâce à un mécanisme d'attention. Dans un cadre de transfert d'expressivité en mode parallèle, cette technique se révèle efficace (meilleur score MOS expressivité parmi les approches autorégressives étudiées).

Nous avons ensuite étudié le transfert d'expressivité avec des systèmes de synthèse de

parole non-autorégressifs mettant en oeuvre des décodeurs reposant respectivement sur Glow (*Generative Flow*) et sur des modèles probabilistes de diffusion (DPM). Là aussi, nous avons étudié plusieurs variantes d'encodeur de l'expressivité, la plus performante s'inspire des systèmes de reconnaissance d'émotions, et exploite des caratéristiques acoustiques à différentes échelles pour déterminer le plongement d'expressivité. Dans le contexte de transfert d'expressivité, les expériences ont montré un léger avantage de l'utilisation du décodeur Glow par rapport à DPM. Pour le transfert d'expressivité, en comparaison des approches autorégressives précédentes, les approches non-autorégressives conduisent à des scores MOS locuteur meilleurs, mais au détriment de scores MOS expressivité un peu plus bas.

Deux aspects sont importants à considérer dans les travaux futures : les performances globales des systèmes de synthèse de la parole et la capacité du modèle à transférer l'expressivité. L'utilisation d'un convertisseur graphème-phonème or pseudo-phonème entraîné conjointement avec les systèmes de synthèse de parole de bout en bout peut renforcer l'information phonetique et sa cohérence dans le système. En outre, une approche globale devrait aider un système de bout en bout a aussi prendre en considération les liens entre expressivité et phonétisation. Certains systèmes de synthèse ont été mis en oeuvre avec des vocodeurs pré-entrainés sur des données anglaises ; c'est le cas de Wavglow utilisé pour les systèmes autorégressifs de bout en bout à grain fin, et de Hifi-GAN utilisé pour les systèmes non-autorégressifs de bout en bout. Il serait intéressant d'évaluer l'impact sur les performances de synthèse expressive d'un apprentissage de ces vocodeurs sur des données françaises.

Les performances des systèmes de synthèse de parole de bout en bout dépendent fortement de la quantité de données d'entraînement disponibles, y compris pour la parole expressive. Il est également important de bien déméler les plongements locuteur et expressivité pour que chacun représente l'information voulue uniquement. L'utilisation de plusieurs corpus de parole expressive devrait aider à cela. Tout au long de la thèse, nous avons développé des systèmes de parole expressive sans tenir compte de la sémantique et du contenu textuel. Il sera intéressant d'analyser l'effet du contenu textuel sur la perception de la synthèse vocale expressive.

Bibliography

[Adigwe et al., 2018] Adigwe, A., Tits, N., Haddad, K. E., Ostadabbas, S., and Dutoit, T. (2018). The emotional voices database: Towards controlling the emotion dimension in voice generation systems. *proceedings of International Conference on Statistical Language and Speech Processing (SLSP)*.

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

[Akhulkova et al., 2021] Akhulkova, Y., Hickey, S., and García, B. A. (2021). Nimdzi language technology atlas: The definitive guide to the language technology landscape. In *https://www.nimdzi.com/language-technology-atlas/*.

[Akuzawa et al., 2018] Akuzawa, K., Iwasawa, Y., and Matsuo, Y. (2018). Expressive speech synthesis via modeling expressions with variational autoencoder. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Allen et al., 1987] Allen, J., Hunnicutt, M. S., Klatt, D. H., Armstrong, R. C., and Pisoni, D. B. (1987). From text to speech: the mitalk system. volume 81.

[Allport and Vernon, 1933] Allport, G. W. and Vernon, P. E. (1933). Studies in expressive movement. *Macmillan Publishers*.

[Amodei et al., 2016] Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., Chen, J., Chen, J., Chen, Z., Chrzanowski, M., Coates, A., Diamos, G., Ding, K., Du, N., Elsen, E., Engel, J., Fang, W., Fan, L., Fougner, C., Gao, L., Gong, C., Hannun, A., Han, T., Johannes, L. V., Jiang, B., Ju, C., Jun, B., LeGresley, P., Lin, L., Liu, J., Liu, Y., Li, W., Li, X., Ma, D., Narang, S., Ng, A., Ozair, S., Peng, Y., Prenger, R., Qian, S., Quan, Z., Raiman, J., Rao, V., Satheesh, S., Seetapun, D., Sengupta, S., Srinet, K., Sriram, A., Tang, H., Tang, L., Wang, C., Wang, J., Wang, K., Wang, Y., Wang, Z., Wang, Z., Wu, S., Wei, L., Xiao, B., Xie, W., Xie, Y., Yogatama, D., Yuan, B., Zhan, J., and Zhu, Z. (2016). Deep speech 2: End-to-end speech recognition in English and Mandarin. In *proceedings of International Conference on Machine Learning (ICML)*.

[An et al., 2017] An, S., Ling, Z., and Dai, L. (2017). Emotional statistical parametric speech synthesis using lstm-rnns. *In proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*.

[Anderson, 1982] Anderson, B. (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*.

[Apicella et al., 2021] Apicella, A., Donnarumma, F., Isgrò, F., and Prevete, R. (2021). A survey on modern trainable activation functions. *Journal of the International Neural Network Society*, 138:14–32.

[Aranguren and Tonnelat, 2014] Aranguren, M. and Tonnelat, S. (2014). Emotional transactions in the paris subway: Combining naturalistic videotaping, objective facial coding and sequential analysis in the study of nonverbal emotional behavior. *Journal of Nonverbal Behavior*, 38:495–521.

[Arias-Vergara et al., 2021] Arias-Vergara, T., Arias-Vergara, T., Arias-Vergara, T., Klumpp, P., Vásquez-Correa, J. C., Vásquez-Correa, J. C., Nöth, E., Orozco-Arroyave, J. R., Orozco-Arroyave, J. R., and Schuster, M. (2021). Multi-channel spectrograms for speech processing applications using deep learning methods. *Pattern Analysis Application*, 24:423–431.

[Arias-Vergara et al., 2017] Arias-Vergara, T., Vásquez-Correa, J. C., and Orozco-Arroyave, J. R. (2017). Parkinson's disease and aging: Analysis of their effect in phonation and articulation of speech. *Cognitive Computation*, 9:731–748.

[Arik et al., 2017] Arik, S. Ö., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Ng, A., Raiman, J., Sengupta, S., and Shoeybi, M. (2017). Deep voice: Real-time neural text-to-speech. In *proceedings of International Conference on Machine Learning (ICML)*.

[Ashfahani et al., 2020] Ashfahani, A., Pratama, M., Lughofer, E., and Ong, Y. S. (2020). Devdan: Deep evolving denoising autoencoder. *Journal of Neurocomputing*, 390:297–314.

[Ba et al., 2016] Ba, J., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *ArXiv*. `https://arxiv.org/pdf/1607.06450.pdf`.

[Baby et al., 2020] Baby, A., Vinnaitherthan, S., Adiga, N., Jawale, P., Badam, S., Adavanne, S., and Konjeti, S. (2020). An ASR guided speech intelligibility measure for TTS model selection. *ArXiv*. `https://arxiv.org/abs/2006.01463`.

[Bai et al., 2020] Bai, Z., Zhang, X.-L., and Chen, J. (2020). Cosine metric learning based speaker verification. *Speech Communication*, 118:10–20.

[Barra-Chicote et al., 2010] Barra-Chicote, R., Yamagishi, J., King, S., Montero-Martínez, J. M., and Macias-Guarasa, J. (2010). Analysis of statistical parametric and unit selection speech synthesis systems applied to emotional speech. *Speech Commun.*, 52:394–404.

[Beliaev and Ginsburg, 2021] Beliaev, S. and Ginsburg, B. (2021). Talknet 2: Non-autoregressive depth-wise separable convolutional model for speech synthesis with explicit pitch and duration prediction. *ArXiv*. `https://arxiv.org/pdf/2104.08189.pdf`.

[Beliaev et al., 2020] Beliaev, S., Rebryk, Y., and Ginsburg, B. (2020). Talknet: Fully-convolutional non-autoregressive speech synthesis model. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Beller, 2010] Beller, G. (2010). Expresso : Transformation of expressivity in speech. In *proceedings of Speech Prosody Conference*.

[Beller et al., 2006] Beller, G., Hueber, T., Schwarz, D., and Rodet, X. (2006). Speech rates in French expressive speech. In *proceedings of Speech Prosody Conference*.

[Bengio, 2012] Bengio, Y. (2012). Deep learning of representations for unsupervised and transfer learning. In *proceedings of International Conference on Machine Learning (ICML)*.

[Bengio et al., 2013] Bengio, Y., Courville, A. C., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:1798–1828.

[Besson et al., 2002] Besson, M., Magne, C. L., and Schön, D. (2002). Emotional prosody: sex differences in sensitivity to speech melody. *Trends in Cognitive Sciences*, 6.

[Bishop, 2006] Bishop, C. M. (2006). Pattern recognition and machine learning. Springer.

[Bishop and Nasrabadi, 2007] Bishop, C. M. and Nasrabadi, N. M. (2007). Pattern recognition and machine learning. *Springer*.

[Boulila et al., 2021] Boulila, W., Driss, M., Al-Sarem, M., Saeed, F., and Krichen, M. (2021). Weight initialization techniques for deep learning algorithms in remote sensing: Recent trends and future perspectives. *ArXiv*. `https://arxiv.org/pdf/2102.07004.pdf`.

[Bowman et al., 2016] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Józefowicz, R., and Bengio, S. (2016). Generating sentences from a continuous space. *In proceedings of SIGNLL Conference on Computational Natural Language Learning (CoNLL)*.

[Broad et al., 2021] Broad, T., Leymarie, F. F., and Grierson, M. (2021). Network bending: Expressive manipulation of deep generative models. In *proceedings of International Conference on Artificial Intelligence in Music, Sound, Art and Design (EvoMUSART)*.

[Brognaux et al., 2012] Brognaux, S., Roekhaut, S., Drugman, T., and Beaufort, R. (2012). Train and align: A new online tool for automatic phonetic alignment. *In proceedings of IEEE Spoken Language Technology Workshop (SLT)*.

[Bromley et al., 1993] Bromley, J., Bentz, J. W., Bottou, L., Guyon, I., LeCun, Y. A., Moore, C., Säckinger, E., and Shah, R. (1993). Signature verification using a "siamese" time delay neural network. In *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Bruns et al., 2021] Bruns, M., Ossevoort, S., and Petersen, M. G. (2021). Expressivity in interaction: a framework for design. *In proceedings of Conference on Human Factors in Computing Systems*.

[Buck et al., 1972] Buck, R., Savin, V., Miller, R., and Caul, W. F. (1972). Communication of affect through facial expressions in humans. *Journal of personality and social psychology*.

[Burkhardt et al., 2005] Burkhardt, F., Paeschke, A., Rolfes, M., Sendlmeier, W. F., and Weiss, B. (2005). A database of German emotional speech. *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Burkhardt and Sendlmeier, 2000] Burkhardt, F. and Sendlmeier, W. F. (2000). Verification of acoustical correlates of emotional speech using formant-synthesis. In *proceedings of the ISCA Workshop on Speech and Emotion*.

[Cahn, 1990] Cahn, J. (1990). The generation of affect in synthesized speech. *Journal of the American Voice I/O Society*, 8.

[Campbell, 2005] Campbell, N. (2005). Developments in corpus-based speech synthesis: Approaching natural conversational speech. *IEICE transactions on information and systems*, 88-D:376–383.

[Campbell and Black, 1997] Campbell, N. and Black, A. W. (1997). Prosody and the selection of source units for concatenative synthesis. *Progress in Speech Synthesis*.

[Casanova et al., 2021] Casanova, E., Shulby, C., Gölge, E., Müller, N. M., de Oliveira, F. S., Candido, A., Soares, A. S., Aluísio, S. M., and Ponti, M. (2021). Sc-glowtts: an efficient zero-shot multi-speaker text-to-speech model. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Cawley and Noakes, 1993] Cawley, G. C. and Noakes, P. D. (1993). LSP speech synthesis using backpropagation networks. *In proceedings of International Conference on Artificial Neural Networks (ICANN)*.

[Cerisara et al., 2009] Cerisara, C., Mella, O., and Fohr, D. (2009). Jtrans: an open-source software for semi-automatic text-to-speech alignment. *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Charfuelan and Steiner, 2013] Charfuelan, M. and Steiner, I. (2013). Expressive speech synthesis in mary TTS using audiobook data and emotion ml. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Chaudhari et al., 2019] Chaudhari, S., Polatkan, G., Ramanath, R., and Mithal, V. (2019). An attentive survey of attention models. `https://arxiv.org/pdf/1904.02874.pdf`.

[Chen and Braunschweiler, 2013] Chen, L. and Braunschweiler, N. (2013). Unsupervised speaker and expression factorization for multi-speaker expressive synthesis of ebooks. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Chen et al., 2015] Chen, L., Braunschweiler, N., and Gales, M. J. F. (2015). Speaker and expression factorization for audiobook data: Expressiveness and transplantation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(4):605–618.

[Chen et al., 2021a] Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., and Chan, W. (2021a). Wavegrad: Estimating gradients for waveform generation. In *proceedings of International Conference on Learning Representations (ICLR)*.

[Chen et al., 2021b] Chen, N., Zhang, Y., Zen, H., Weiss, R. J., Norouzi, M., Dehak, N., and Chan, W. (2021b). Wavegrad 2: Iterative refinement for text-to-speech synthesis. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Chen et al., 2018] Chen, T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural ordinary differential equations. In *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Chen et al., 2019] Chen, X., He, L., Xu, C., and Liu, J. (2019). Distance-dependent metric learning. *IEEE Signal Processing Letters*, 26.

[Cho et al., 2014] Cho, K., van Merrienboer, B., Çaglar Gülçehre, Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

[Chorowski et al., 2015] Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. In *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Chung et al., 2018] Chung, J. S., Nagrani, A., and Zisserman, A. (2018). Voxceleb2: Deep speaker recognition. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Dael et al., 2012] Dael, N., Mortillaro, M., and Scherer, K. (2012). Emotion expression in body action and posture. *Emotion*.

[Dahmani et al., 2019] Dahmani, S., Colotte, V., Girard, V., and Ouni, S. (2019). Conditional variational auto-encoder for text-driven expressive audio visual speech synthesis. *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Dai et al., 2021] Dai, X., Gong, C., Wang, L., and Zhang, K. (2021). Information sieve: Content leakage reduction in end-to-end prosody for expressive speech synthesis. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Dalgleish and Power, 1999] Dalgleish, T. and Power, M. (1999). Handbook of cognition and emotion.

[Dall et al., 2014] Dall, R., Yamagishi, J., and King, S. (2014). Rating naturalness in speech synthesis: The effect of style and expectation. *In proceedings of Speech Prosody Conference*.

[Dalsgaard et al., 2016] Dalsgaard, P., Halskov, K., and Iversen, O. (2016). Participation Gestalt: Analysing participatory qualities of interaction in public space. *In proceedings of Conference on Human Factors in Computing Systems*.

[Dalsgård and Hansen, 2008] Dalsgård, P. and Hansen, L. K. (2008). Performing perception—staging aesthetics of interaction. *ACM Transactions on Computer-Human Interaction*, 15:13:1–13:33.

[Darryl, 2018] Darryl, S. (2018). Garbage in: Garbage out. *Gale Academic One File, Quality*, 57.

[Darwin, 1872] Darwin, C. (1872). The expression of emotions in man and animals. In *London: John Murray. 1st edition*.

[Davis et al., 2020] Davis, S., Coccia, G., Gooch, S., and Mack, J. (2020). Empirical evaluation of deep learning model compression techniques on the wavenet vocoder. *ArXiv*. `https://arxiv.org/pdf/2011.10469.pdf`.

[Dehak et al., 2007] Dehak, N., Dumouchel, P., and Kenny, P. (2007). Modeling prosodic features with joint factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7).

[Deller and Hansen, 2005] Deller, J. and Hansen, J. (2005). Methods, models, and algorithms for modern speech processing. *The Electrical Engineering Handbook*.

[den Berg et al., 2018] den Berg, R. V., Hasenclever, L., Tomczak, J. M., and Welling, M. (2018). Sylvester normalizing flows for variational inference. In *proceedings of Conference on Uncertainty in Artificial Intelligence (UAI)*.

[Deshmukh and Espy-Wilson, 2003] Deshmukh, O. and Espy-Wilson, C. (2003). A measure of aperiodicity and periodicity in speech. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Dilokthanakul et al., 2016] Dilokthanakul, N., Mediano, P. A. M., Garnelo, M., Lee, M. J., Salimbeni, H., Arulkumaran, K., and Shanahan, M. (2016). Deep unsupervised clustering with Gaussian mixture variational autoencoders. *In proceedings of International Conference on Computer Vision (ICCV)*.

[Dinh et al., 2015] Dinh, L., Krueger, D., and Bengio, Y. (2015). NICE: Non-linear independent components estimation. *ArXiv*. `https://arxiv.org/pdf/1410.8516.pdf`.

[Dinh et al., 2017] Dinh, L., Sohl-Dickstein, J., and Bengio, S. (2017). Density estimation using Real NVP. *In proceedings of International Conference on Learning Representations (ICLR)*.

[Donahue et al., 2019] Donahue, C., McAuley, J., and Puckette, M. (2019). Adversarial audio synthesis. In *proceedings of International Conference on Learning Representations (ICLR)*.

[Douros, 2020] Douros, I. K. (2020). *Towards a 3 dimensional dynamic generic speaker model to study geometry simplifications of the vocal tract using magnetic resonance imaging data*. PhD Theses, Université de Lorraine.

[Ekman, 1992] Ekman, P. (1992). An argument for basic emotions. *In proceedings of Cognition & Emotion*.

[Ekman, 1999] Ekman, P. (1999). Basic emotions. *Handbook of cognition and emotion*, 99-3:550–553.

[Ekman et al., 1969] Ekman, P., Sorenson, E., and Friesen, W. (1969). Pan-cultural elements in facial displays of emotion. *Science*, 164:86 – 88.

[El-Kaddoury et al., 2019] El-Kaddoury, M., Mahmoudi, A., and Himmi, M. M. (2019). Deep generative models for image generation: A practical comparison between variational autoencoders and generative adversarial networks. In *proceedings of Mobile, Secure, and Programmable Networking*.

[Elias et al., 2021a] Elias, I., Zen, H., Shen, J., Zhang, Y., Jia, Y., Weiss, R. J., and Wu, Y. (2021a). Parallel Tacotron: Non-autoregressive and controllable TTS. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Elias et al., 2021b] Elias, I., Zen, H., Shen, J., Zhang, Y., Ye, J., Skerry-Ryan, R., and Wu, Y. (2021b). Parallel Tacotron 2: A non-autoregressive neural TTS model with differentiable duration modeling. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Elie and Laprie, 2016] Elie, B. and Laprie, Y. (2016). Extension of the single-matrix formulation of the vocal tract: Consideration of bilateral channels and connection of self-oscillating models of the vocal folds with a glottal chink. *Speech Communication*, 82:85–96.

[Eminağaoğlu and Gökşen, 2020] Eminağaoğlu, M. and Gökşen, Y. (2020). A new similarity measure for document classification and text mining. *KnE Social Sciences*, 12:353–366.

[Esling et al., 2019] Esling, P., Masuda, N., Bardet, A., Despres, R., and Chemla-Romeu-Santos, A. (2019). Flow synthesizer: Universal audio synthesizer control with normalizing flows. *Applied Sciences*, 10.

[Fairbanks and Pronovost, 1939a] Fairbanks, G. and Pronovost, W. (1939a). An experimental study of the pitch characteristics of the voice during the expression of emotion. *Communication Monographs*, 6:87–104.

[Fairbanks and Pronovost, 1939b] Fairbanks, G. and Pronovost, W. L. (1939b). An experimental study of the pitch characteristics of the voice during the expression of emotion. *Communication Monographs*, 6.

[Fan et al., 2014] Fan, Y., Qian, Y., Xie, F.-L., and Soong, F. K. (2014). TTS synthesis with bidirectional LSTM based recurrent neural networks. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Fernandez and Ramabhadran, 2007] Fernandez, R. and Ramabhadran, B. (2007). Automatic exploration of corpus-specific properties for expressive text-to-speech: a case study in emphasis. *In proceedings of International Workshop on Semantic Search Over the Web (SSW)*.

[Flanagan, 1971] Flanagan, J. L. (1971). Speech analysis, synthesis and perception. In *Springer*.

[Flanagan and Rabiner, 1973] Flanagan, J. L. and Rabiner, L. R. (1973). Speech synthesis.

[Flavio et al., 2011] Flavio, P. R., Florencio, D., Zhang, C., and Seltzer, M. (2011). CROWDMOS: An approach for crowdsourcing mean opinion score studies. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Friedman, 1979] Friedman, H. S. (1979). The concept of skill in nonverbal communication: Implications for understanding social interaction. *Skill in nonverbal communication*.

[Fukushima, 1980] Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*.

[Gatys et al., 2016] Gatys, L. A., Ecker, A. S., and Bethge, M. (2016). Image style transfer using convolutional neural networks. *In proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[Gers and Schmidhuber, 2000] Gers, F. A. and Schmidhuber, J. (2000). Recurrent nets that time and count. *In proceedings of International Joint Conference on Neural Networks (IJCNN).*

[Gers et al., 2000] Gers, F. A., Schmidhuber, J., and Cummins, F. A. (2000). Learning to forget: Continual prediction with lstm. *Neural Computation.*

[Gibiansky et al., 2017] Gibiansky, A., Arik, S. Ö., Diamos, G., Miller, J., Peng, K., Ping, W., Raiman, J., and Zhou, Y. (2017). Deep voice 2: Multi-speaker neural text-to-speech. In *proceedings of Conference on Neural Information Processing Systems (NIPS).*

[Glasmachers, 2017] Glasmachers, T. (2017). Limits of end-to-end learning. In *proceedings of Asian Conference on Machine Learning (ACML).*

[Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *proceedings of International Conference on Artificial Intelligence and Statistics (AISTATS).*

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning.* MIT Press. `http://www.deeplearningbook.org`.

[Goodfellow et al., 2014] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial networks. *In proceedings of Conference on Neural Information Processing Systems (NIPS).*

[Gournay et al., 2018] Gournay, P., Lahaie, O., and Lefebvre, R. (2018). A Canadian French emotional speech dataset. *In proceedings of ACM Multimedia Systems Conference.*

[Govind and Prasanna, 2013] Govind, D. and Prasanna, S. R. M. (2013). Expressive speech synthesis: a review. *International Journal of Speech Technology*, 16.

[Grathwohl et al., 2019] Grathwohl, W., Chen, R. T. Q., Bettencourt, J., Sutskever, I., and Duvenaud, D. K. (2019). Ffjord: Free-form continuous dynamics for scalable reversible generative models. *In proceedings of International Conference on Learning Representations (ICLR).*

[Graves et al., 2013] Graves, A., Mohamed, A. R., and Hinton, G. E. (2013). Speech recognition with deep recurrent neural networks. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*

[Griffin and Lim, 1985] Griffin, D. and Lim, J. (1985). A new model-based speech analysis/synthesis system. In *proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*

[Griffin and Lim, 1988] Griffin, D. and Lim, J. (1988). Multiband excitation vocoder. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*

[Hadsell et al., 2006] Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. *In proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:1735–1742.

[Hashimoto et al., 2015] Hashimoto, K., Oura, K., Nankaku, Y., and Tokuda, K. (2015). The effect of neural networks in statistical parametric speech synthesis. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Hashizawa et al., 2004] Hashizawa, Y., Takeda, S., Hamzah, M. D., and Ohyama, G. (2004). On the differences in prosodic features of emotional expressions in japanese speech according to the degree of the emotion. In *proceedings of Speech and Prosody*.

[Hassanien et al., 2008] Hassanien, A. E., Abraham, A., and Kacprzyk, J. (2008). Computational intelligence in multimedia processing: Recent advances. *Springer*.

[Hendrycks and Gimpel, 2016] Hendrycks, D. and Gimpel, K. (2016). Gaussian Error Linear Units (GELUs). *ArXiv*. `https://arxiv.org/pdf/1606.08415.pdf`.

[Hinton and Salakhutdinov, 2006] Hinton, G. E. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *Science*.

[Hinton and Sejnowski, 1983] Hinton, G. E. and Sejnowski, J. (1983). Optimal perceptual inference. In *proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Ho et al., 2020] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *In proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Hochreiter, 1998] Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*.

[Hofer et al., 2005] Hofer, G., Richmond, K., and Clark, R. A. J. (2005). Informed blending of databases for emotional speech synthesis. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Hoffer and Ailon, 2015] Hoffer, E. and Ailon, N. (2015). Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition (SIMBAD)*.

[Hojo et al., 2016] Hojo, N., Ijima, Y., and Mizuno, H. (2016). An investigation of DNN-based speech synthesis using speaker codes. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Honnet et al., 2017] Honnet, P.-E., Lazaridis, A., Garner, P. N., and Yamagishi, J. (2017). The Siwis French speech synthesis database design and recording of a high quality french database for speech synthesis. `https://datashare.ed.ac.uk/handle/10283/2353`.

[Hook et al., 2011] Hook, J., Green, D., McCarthy, J. C., Taylor, S., Wright, P., and Olivier, P. (2011). A VJ centered exploration of expressive interaction. *In proceedings of the Conference on Human Factors in Computing Systems*.

[Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *National Academy of Sciences of the United States of America.*

[Hsu et al., 2019] Hsu, W.-N., Zhang, Y., Weiss, R. J., Zen, H., Wu, Y., Wang, Y., Cao, Y., Jia, Y., Chen, Z., Shen, J., Nguyen, P., and Pang, R. (2019). Hierarchical generative modeling for controllable speech synthesis. *ArXiv.* `https://arxiv.org/pdf/1810.07217.pdf`.

[Hu et al., 2014] Hu, J., Lu, J., and Tan, Y.-P. (2014). Discriminative deep metric learning for face verification in the wild. *In proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[Hu et al., 2020] Hu, T. Y., Shrivastava, A., Tuzel, O., and Dhir, C. S. (2020). Unsupervised style and content separation by minimizing mutual information for speech synthesis. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*

[Huang et al., 2018] Huang, W., Ding, H., and Chen, G. (2018). A novel deep multichannel residual networks-based metric learning method for moving human localization in video surveillance. *Signal Processing*, 142.

[Hubel and Wiesel, 1968] Hubel, D. H. and Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *Journal of Physiology.*

[Iida et al., 2000] Iida, A., Campbell, N., Iga, S., Higuchi, F., and Yasumura, M. (2000). A speech synthesis system with emotion for assisting communication.

[India et al., 2019] India, M., Safari, P., and Hernando, J. (2019). Self multi-head attention for speaker recognition. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH).*

[Inoue et al., 2017] Inoue, K., Hara, S., Abe, M., Hojo, N., and Ijima, Y. (2017). An investigation to transplant emotional expressions in DNN-based TTS synthesis. *In proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC).*

[Ioffe and Szegedy, 2015] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *In proceedings of International Conference on Machine Learning (ICML).*

[Ishi and Campbell, 2004] Ishi, C. and Campbell, N. (2004). Analysis of acoustic-prosodic features of spontaneous expressive speech. *Revista de Estudos da Linguagem.*

[Ito and Johnson, 2017] Ito, K. and Johnson, L. (2017). The LJSpeech dataset. *Link: https://keithito.com/LJ-Speech-Dataset/.*

[Jeong et al., 2021] Jeong, M., Kim, H., Cheon, S. J., Choi, B. J., and Kim, N. S. (2021). Diff-TTS: A denoising diffusion model for text-to-speech. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH).*

[Jeong et al., 2018] Jeong, Y., Lee, S., Park, D.-G., and Park, K.-H. (2018). Accurate age estimation using multi-task siamese network-based deep metric learning for frontal face images. *Symmetry*, 10:385.

[Jia et al., 2021] Jia, Y., Ramanovich, M. T., Remez, T., and Pomerantz, R. (2021). Translatotron 2: Robust direct speech-to-speech translation. *ArXiv*. `https://arxiv.org/pdf/2107.08661.pdf`.

[Jia et al., 2018] Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J., Ren, F., Chen, z., Nguyen, P., Pang, R., Lopez Moreno, I., and Wu, Y. (2018). Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Jin et al., 2018] Jin, Z., Finkelstein, A., Mysore, G. J., and Lu, J. (2018). FFTnet: A real-time speaker-dependent neural vocoder. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Jing et al., 2020] Jing, Y., Yang, Y., Feng, Z., Ye, J., and Song, M. (2020). Neural style transfer: A review. *IEEE Transactions on Visualization and Computer Graphics*.

[Johnstone and Scherer, 1999] Johnstone, T. and Scherer, K. (1999). The effects of emotions on voice quality. *proceedings of International Congress of Phonetic Sciences (ICPhS)*.

[Jordan and Mitchell, 2015] Jordan, M. I. and Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. Science.

[Joshi et al., 2015] Joshi, A., Kale, S., Chandel, S., and Pal, D. K. (2015). Likert scale: Explored and explained. *British Journal of Applied Science and Technology*, 7:396–403.

[Kalchbrenner et al., 2018] Kalchbrenner, N., Elsen, E., Simonyan, K., Noury, S., Casagrande, N., Lockhart, E., Stimberg, F., van den Oord, A., Dieleman, S., and Kavukcuoglu, K. (2018). Efficient neural audio synthesis. *Machine Learning Research (MLR)*.

[Karaali et al., 1998] Karaali, O., Corrigan, G., Massey, N., Miller, C., Schnurr, O., and Mackie, A. (1998). A high quality text-to-speech system composed of multiple neural networks. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Karlapati et al., 2020] Karlapati, S., Moinet, A., Joly, A., Klimkov, V., S'aez-Trigueros, D., and Drugman, T. (2020). Copycat: Many-to-many fine-grained prosody transfer for neural text-to-speech. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Kawahara et al., 2001] Kawahara, H., Estill, J., and Fujimura, O. (2001). Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system straight. *In proceedings of International Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications (MAVEBA)*.

[Kaya and Bilge, 2019] Kaya, M. and Bilge, H. S. (2019). Deep metric learning: A survey. *Symmetry*, 11.

[Kearney et al., 2009] Kearney, G., Masterson, C., Adams, S., and Boland, F. (2009). Dynamic time warping for acoustic response interpolation: Possibilities and limitations. *In proceedings of European Signal Processing Conference (EUSIPCO)*.

[Kim et al., 2020] Kim, J., Kim, S., Kong, J., and Yoon, S. (2020). Glow-TTS: A generative Flow for text-to-speech via monotonic alignment search. *In proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Kim et al., 2021] Kim, J., Shim, H., Jung, J., and Yu, H. (2021). Learning metrics from mean teacher: A supervised learning method for improving the generalization of speaker verification system. *ArXiv*. `https://arxiv.org/abs/2104.06604`.

[Kim et al., 2019] Kim, S., gil Lee, S., Song, J., Kim, J., and Yoon, S. (2019). Flowavenet : A generative flow for raw audio. *In proceedings of International Conference on Machine Learning (ICML)*.

[Kim et al., 2004] Kim, Y.-J., Syrdal, A., and Jilka, M. (2004). Improving TTS by higher agreement between predicted versus observed pronunciations. In *proceedings of International Workshop on Semantic Search Over the Web (SSW)*.

[Kinchella and Guo, 2021] Kinchella, J. and Guo, K. (2021). Facial expression ambiguity and face image quality affect differently on expression interpretation bias. *Perception*.

[King, 2010] King, S. (2010). A tutorial on HMM speech synthesis. *In Sadhana - Academy Proceedings in Engineering Sciences, Indian Institute of Sciences*.

[King, 2011] King, S. (2011). An introduction to statistical parametric speech synthesis. *Sadhana India*, 36:837–852.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *In proceedings of International Conference on Learning Representations (ICLR)*.

[Kingma and Dhariwal, 2018] Kingma, D. P. and Dhariwal, P. (2018). Glow: Generative flow with invertible 1x1 convolutions. In *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Kingma et al., 2014] Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. (2014). Semi-supervised learning with deep generative models. In *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Kingma et al., 2016] Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Kingma and Welling, 2014] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. *proceedings of International Conference on Learning Representations (ICLR)*.

[Kinsler et al., 1999] Kinsler, L. E., Frey, A. R., Coppens, A. B., and Sanders, J. V. (1999). Fundamentals of acoustics. John Wiley and Sons.

[Klambauer et al., 2017] Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. (2017). Self-normalizing neural networks. In *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Klatt, 1980] Klatt, D. H. (1980). Software for a cascade/parallel formant synthesizer. *Journal of the Acoustical Society of America*, 67:1110–1121.

[Klatt, 1987] Klatt, D. H. (1987). Review of text-to-speech conversion for English. *The Journal of the Acoustical Society of America*, 82:737–93.

[Klejsa et al., 2019] Klejsa, J., Hedelin, P., Zhou, C., Fejgin, R., and Villemoes, L. F. (2019). High-quality speech coding with sample RNN. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Klimkov et al., 2019] Klimkov, V., Ronanki, S., Rohnke, J., and Drugman, T. (2019). Fine-grained robust prosody transfer for single-speaker neural text-to-speech. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Kloeden and Platen, 1977] Kloeden, P. E. and Platen, E. (1977). Numerical solution of stochastic differential equations. In *Applications of Mathematics book series*, volume 23.

[Kondo, 2018] Kondo, K. (2018). Subjective quality measurement of speech. *Signals and Communication Technology book series*.

[Kong et al., 2020] Kong, J., Kim, J., and Bae, J. (2020). HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Kong et al., 2021] Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. (2021). Diffwave: A versatile diffusion model for audio synthesis. *ArXiv*. `https://arxiv.org/pdf/2009.09761.pdf`.

[Kreyssig et al., 2018] Kreyssig, F. L., Zhang, C., and Woodland, P. C. (2018). Improved TDNNs using deep kernels and frequency dependent grid-RNNs. *proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Kulkarni et al., 2020a] Kulkarni, A., Colotte, V., and Jouvet, D. (2020a). Deep variational metric learning for transfer of expressivity in multispeaker text to speech. *In proceedings of International Conference on Statistical Language and Speech Processing (SLSP)*.

[Kulkarni et al., 2020b] Kulkarni, A., Colotte, V., and Jouvet, D. (2020b). Transfer learning of the expressivity using FLOW metric learning in multispeaker text-to-speech synthesis. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Kulkarni et al., 2020c] Kulkarni, A., Douros, I. K., Colotte, V., and Jouvet, D. (2020c). Emotion recognition from phoneme-duration information. *In proceedings of International Seminar on Speech Production*.

[Larsen et al., 2016] Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. (2016). Autoencoding beyond pixels using a learned similarity metric. *In proceedings of International Conference on Machine Learning (ICML)*.

[LeCun et al., 1989] LeCun, Y. A., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*.

[LeCun et al., 1998] LeCun, Y. A., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *IEEE*, 38.

[Lee, 2021] Lee, K. (2021). Expressive fastspeech 2. *GitHub repository: https://github.com/keonlee9420/Expressive-FastSpeech2*.

[Lee et al., 2021] Lee, K., Park, K., and Kim, D. (2021). Styler: Style modeling with rapidity and robustness via speechdecomposition for expressive and controllable neural text to speech. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Lewis and Hardzinski, 2015] Lewis, J. R. and Hardzinski, M. L. (2015). Investigating the psychometric properties of the speech user interface service quality questionnaire. *International Journal of Speech Technology*, 18:479–487.

[Li et al., 2018] Li, H., Kang, Y., and Wang, Z. (2018). Emphasis: An emotional phoneme-based acoustic model for speech synthesis system. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Li et al., 2019] Li, N., Liu, S., Liu, Y., Zhao, S., and Liu, M. (2019). Neural speech synthesis with transformer network. In *proceedings of Association for the Advancement of Artificial Intelligence (AAAI)*.

[Li et al., 2021a] Li, T., Wang, X., Xie, Q., Wang, Z., and Xie, L. (2021a). Controllable cross-speaker emotion transfer for end-to-end speech synthesis. *In proceedings of International Symposium on Chinese Spoken Language Processing (ISCSLP)*.

[Li et al., 2021b] Li, X., Song, C., Li, J., Wu, Z., Jia, J., and Meng, H. (2021b). Towards multi-scale style control for expressive speech synthesis. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Liberman et al., 1959] Liberman, A. M., Ingemann, F. J., Lisker, L., and Cooper, F. S. (1959). Minimal rules for synthesizing speech. *Journal of the Acoustical Society of America*, 31:1490–1499.

[Lin et al., 2020] Lin, T., Jin, C., and Jordan, M. I. (2020). On gradient descent ascent for nonconvex-concave minimax problems. *In proceedings of Machine Learning Research (MLR)*.

[Litjens et al., 2017] Litjens, G. J. S., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J., van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42.

[Liu et al., 2015] Liu, J., Deng, Y., Bai, T., and Huang, C. (2015). Targeting ultimate accuracy: Face recognition via deep embedding. *ArXiv*. `https://arxiv.org/pdf/1506.07310.pdf`.

[Liu et al., 2016] Liu, Q., Lee, J., and Jordan, M. I. (2016). A kernelized stein discrepancy for goodness-of-fit tests. In *proceedings of International Conference on Machine Learning (ICML)*.

[Liu et al., 2021] Liu, R., Sisman, B., and Li, H. (2021). Reinforcement learning for emotional text-to-speech synthesis with improved emotion discriminability. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Lu et al., 2013] Lu, H., King, S., and Watts, O. (2013). Combining a vector space representation of linguistic context with a deep neural network for text-to-speech synthesis. *In proceedings of International Workshop on Semantic Search Over the Web (SSW)*.

[Lu et al., 2017] Lu, R., Wu, K., Duan, Z., and Zhang, C. (2017). Deep ranking: Triplet matchnet for music metric learning. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Luan et al., 2017] Luan, F., Paris, S., Shechtman, E., and Bala, K. (2017). Deep photo style transfer. *In proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Luo et al., 2019] Luo, Y., Ceolini, E., Han, C., Liu, S.-C., and Mesgarani, N. (2019). Fasnet: Low-latency adaptive beamforming for multi-microphone audio processing. *In proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

[Luo and Mesgarani, 2019] Luo, Y. and Mesgarani, N. (2019). Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27:1256–1266.

[Masuko et al., 2004] Masuko, T., Kobayashi, T., and Miyanaga, K. (2004). A style control technique for HMM-based speech synthesis. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Matsugu et al., 2003] Matsugu, M., Mori, K., Mitari, Y., and Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural networks : the official journal of the International Neural Network Society*.

[McCulloch and Pitts, 1990] McCulloch, W. S. and Pitts, W. (1990). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*.

[McFee et al., 2015] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., and Nieto, O. (2015). Librosa: Audio and music signal analysis in python. *In proceedings of python in science conference*.

[Miao et al., 2020] Miao, C., Liang, S., Chen, M., Ma, J., Wang, S., and Xiao, J. (2020). Flow-TTS: A non-autoregressive network for text to speech based on flow. In *proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Mirowski et al., 2017] Mirowski, P., Pascanu, R., Viola, F., Soyer, H., Ballard, A., Banino, A., Denil, M., Goroshin, R., Sifre, L., Kavukcuoglu, K., Kumaran, D., and Hadsell, R. (2017). Learning to navigate in complex environments. *In proceedings of International Workshop on Multimodal Understanding and Learning for Embodied Applications*.

[Mohan et al., 2021] Mohan, D., Hu, Q., Teh, T. H., Torresquintero, A., Wallis, C., Staib, M., Foglianti, L., Gao, J., and King, S. (2021). Ctrl-p: Temporal control of prosodic variation for speech synthesis. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Montero et al., 1999] Montero, J. M., Gutiérrez-Arriola, J. M., and Colás, J. (1999). Analysis and modelling of emotional speech in Spanish. In *proceedings of International Congress of Phonetic Sciences (ICPhS)*.

[Moore, 1998] Moore, G. E. (1998). Cramming more components onto integrated circuits. *IEEE*.

[Moore, 1984] Moore, S. (1984). The Stanislavski system: The professional training of an actor. *Penguin*.

[Morise, 2012] Morise, M. (2012). Platinum: A method to extract excitation signals for voice synthesis system. *Acoustical Science and Technology*, 33.

[Morise, 2015] Morise, M. (2015). Cheaptrick, a spectral envelope estimator for high-quality speech synthesis. volume 67.

[Morise, 2016] Morise, M. (2016). D4c, a band-aperiodicity estimator for high-quality speech synthesis. *Speech Communication*, 84:57–65.

[Morise et al., 2009] Morise, M., Kawahara, H., and Katayose, H. (2009). Fast and reliable f0 estimation method based on the period extraction of vocal fold vibration of singing voice and speech. *In proceedings of International Conference on Audio Engineering Society*.

[Morise et al., 2016] Morise, M., Yolomori, F., and Ozawa, K. (2016). WORLD: A vocoder-based high-quality speech synthesis system for real-time applications. *IEICE Transactions on Information and Systems*, 99-D:1877–1884.

[Muller, 2007] Muller, M. (2007). Dynamic time warping. *Information retrieval for music and motion*, 69.

[Mullin and Rosenblatt, 1962] Mullin, A. A. and Rosenblatt, F. (1962). Principles of neurodynamics. *Springer*.

[Murphy, 2012] Murphy, K. P. (2012). Machine learning - a probabilistic perspective. *Adaptive computation and machine learning series*.

[Murray and Arnott, 1993] Murray, I. R. and Arnott, J. L. (1993). Toward the simulation of emotion in synthetic speech: a review of the literature on human vocal emotion. *The Journal of the Acoustical Society of America*, 93.

[Murray and Arnott, 1995] Murray, I. R. and Arnott, J. L. (1995). Implementation and testing of a system for producing emotion-by-rule in synthetic speech. *Speech Communication*, 16.

[Nalisnick et al., 2016] Nalisnick, E. T., Hertel, L., and Smyth, P. (2016). Approximate inference for deep latent Gaussian mixtures. In *NIPS Workshop on Bayesian Deep Learning*.

[Narayanaswamy et al., 2019] Narayanaswamy, V. S., Thiagarajan, J. J., Song, H., and Spanias, A. (2019). Designing an effective metric learning pipeline for speaker diarization. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Nguyen and Bai, 2010] Nguyen, H. V. and Bai, L. (2010). Cosine similarity metric learning for face verification. *In proceedings of the Asian Conference on Computer Vision (ACCV)*.

[Nichol and Dhariwal, 2021] Nichol, A. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. *In proceedings of International Conference on Machine Learning (ICML)*.

[Nose and Kobayashi, 2013] Nose, T. and Kobayashi, T. (2013). An intuitive style control technique in HMM-based expressive speech synthesis using subjective style intensity and multiple-regression global variance model. *Speech Communication*, 55.

[Ohtani et al., 2015] Ohtani, Y., Nasu, Y., Morita, M., and Akamine, M. (2015). Emotional transplant in statistical speech synthesis based on emotion additive model. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Okabe et al., 2018] Okabe, K., Koshinaka, T., and Shinoda, K. (2018). Attentive statistics pooling for deep speaker embedding. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Oliveira et al., 2008] Oliveira, L., Paulo, S., Figueira, L., Mendes, C., Nunes, A., and Godinho, J. (2008). Methodologies for designing and recording speech databases for corpus based synthesis. *In proceedings of International Conference on Language Resources and Evaluation (LREC)*.

[Oord et al., 2016] Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. In *proceedings of International Workshop on Semantic Search Over the Web (SSW)*.

[Orozco-Arroyave et al., 2018] Orozco-Arroyave, J. R., Vásquez-Correa, J. C., Vargas-Bonilla, J. F., Arora, R., Dehak, N., Nidadavolu, P. S., Christensen, H., Rudzicz, F., Yancheva, M., Chinaei, H., Vann, A., Vogler, N., Bocklet, T., Cernak, M., Hannink, J., and Nöth, E. (2018). Neurospeech: An open-source software for parkinson's speech analysis. *In proceedings of International Conference on Digital Signal Processing*.

[O'Shaughnessy, 1987] O'Shaughnessy, D. D. (1987). Speech communication : human and machine. Wiley-IEEE Press.

[Paeschke, 2004] Paeschke, A. (2004). Global trend of fundamental frequency in emotional speech. In *proceedings of Speech and Prosody*.

[Panayotov et al., 2015] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. *In proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

[Park et al., 2018] Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., and Kim, Y. (2018). Data synthesis based on generative adversarial networks. *VLDB Endowment*.

[Parker et al., 2018] Parker, J., Stylianou, Y., and Cipolla, R. (2018). Adaptation of an expressive single speaker deep neural network speech synthesis system. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Paul et al., 2021] Paul, D., Mukherjee, S., Pantazis, Y., and Stylianou, Y. (2021). A Universal Multi-Speaker Multi-Style Text-to-Speech via Disentangled Representation Learning Based on Rényi Divergence Minimization. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Peng et al., 2020] Peng, K., Ping, W., Song, Z., and Zhao, K. (2020). Non-autoregressive neural text-to-speech. In *proceedings of International Conference on Machine Learning (ICML)*.

[Peng et al., 2021] Peng, Z., Lu, Y., Pan, S., and Liu, Y. (2021). Efficient speech emotion recognition using multi-scale CNN and attention. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Ping et al., 2019] Ping, W., Peng, K., and Chen, J. (2019). Clarinet: Parallel wave generation in end-to-end text-to-speech. *In proceedings of International Conference on Learning Representations (ICLR)*.

[Ping et al., 2017] Ping, W., Peng, K., Gibiansky, A., Arik, S. Ö., Kannan, A., Narang, S., Raiman, J., and Miller, J. (2017). Deep voice 3: 2000-speaker neural text-to-speech. *In proceedings of International Conference on Learning Representations (ICLR)*.

[Pitrelli et al., 2006] Pitrelli, J. F., Bakis, R., Eide, E., Fernandez, R., Hamza, W., and Picheny, M. (2006). The IBM expressive text-to-speech synthesis system for American English. *IEEE Transactions on Audio, Speech, and Language Processing*, 14.

[Polkosky and Lewis, 2003] Polkosky, M. D. and Lewis, J. R. (2003). Expanding the MOS: development and psychometric evaluation of the MOS-R and MOS-X. *International Journal of Speech Technology*, 6:161–182.

[Popov et al., 2021] Popov, V., Vovk, I., Gogoryan, V., Sadekova, T., and Kudinov, M. A. (2021). Grad-TTS: A diffusion probabilistic model for text-to-speech. *proceedings of International Conference on Machine Learning (ICML)*.

[Povey et al., 2011] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N. K., Hannemann, M., Motlícek, P., Qian, Y., Schwarz, P., Silovský, J., Stemmer, G., and Veselý, K. (2011). The Kaldi speech recognition toolkit. In *proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

[Prenger et al., 2019] Prenger, R., Valle, R., and Catanzaro, B. (2019). WaveGlow: A Flow-based generative network for speech synthesis. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Qian et al., 2014] Qian, Y., Fan, Y., Hu, W., and Soong, F. K. (2014). On the training aspects of deep neural network (DNN) for parametric TTS synthesis. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Rabiner, 1989] Rabiner, L. (1989). A tutorial on Hidden Markov Models and selected applications in speech recognition. *In proceedings of the IEEE*, 77(2):257–286.

[Rabiner, 1990] Rabiner, L. R. (1990). Readings in speech recognition. *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*.

[Rahim et al., 1993] Rahim, M. G., Goodyear, C. C., Kleijn, W., Schroeter, J., and Sondhi, M. M. (1993). On the use of neural networks in articulatory speech synthesis. *Journal of the Acoustical Society of America*, 93:1109–1121.

[Ren et al., 2020a] Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T.-Y. (2020a). Fastspeech 2: Fast and high-quality end-to-end text to speech. *In proceedings of International Conference on Learning Representations (ICLR)*.

[Ren et al., 2020b] Ren, Y., Liu, J., Tan, X., Zhang, C., Qin, T., Zhao, Z., and Liu, T.-Y. (2020b). SimulSpeech: End-to-end simultaneous speech to text translation. In *proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*.

[Ren et al., 2019] Ren, Y., Ruan, Y., Tan, X., Qin, T., Zhao, S., Zhao, Z., and Liu, T.-Y. (2019). Fastspeech: Fast, robust and controllable text to speech. In *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Rezende and Mohamed, 2015] Rezende, D. J. and Mohamed, S. (2015). Variational inference with normalizing flows. *In proceedings of International Conference on Machine Learning (ICML)*.

[Rezende et al., 2014] Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *proceedings of International Conference on Machine Learning (ICML)*.

[Riedi, 1995] Riedi, M. (1995). A neural-network-based model of segmental duration for speech synthesis. *In proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*.

[Riggio and Riggio, 2002] Riggio, H. R. and Riggio, R. (2002). Emotional expressiveness, extraversion, and neuroticism: A meta-analysis. *Journal of Nonverbal Behavior*.

[Rix et al., 2001] Rix, A., Beerends, J., Hollier, M., and Hekstra, A. (2001). Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Rix et al., 2006] Rix, A., Beerends, J., Kim, D.-S., Kroon, P., and Ghitza, O. (2006). Objective assessment of speech and audio quality—technology and applications. *IEEE Transactions on Audio, Speech, and Language Processing*.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *In proceedings of International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*.

[Rosenblatt, 1958] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *In proceedings of Psychological review*.

[Rumelhart et al., 1986] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*.

[Rumelhart et al., 1987] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1987). Learning internal representations by error propagation. *Parallel distributed processing: explorations in the microstructure of cognition*.

[Russell, 1980] Russell, J. (1980). A circumplex model of affect. *In proceedings of Journal of Personality and Social Psychology*, 39:1161–1178.

[Ruthotto and Haber, 2021] Ruthotto, L. and Haber, E. (2021). An introduction to deep generative modeling. *ArXiv*. `https://arxiv.org/pdf/2103.05180.pdf`.

[Salimans et al., 2015] Salimans, T., Kingma, D. P., and Welling, M. (2015). Markov chain Monte Carlo and variational inference: Bridging the gap. In *proceedings of International Conference on Machine Learning,ICML*.

[Schmidt-Nielsen, 1992] Schmidt-Nielsen, A. (1992). Intelligibility and acceptability testing for speech technology. *Naval research laboratory USA*.

[Schoeffler et al., 2018] Schoeffler, M., Bartoschek, S., Stöter, F., Roess, M., Westphal, S., and Edler, B. (2018). webMUSHRA — a comprehensive framework for web-based listening tests. *Journal of Open Research Software*, 6:8–16.

[Schröder, 2001] Schröder, M. (2001). Emotional speech synthesis: a review. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Schröder, 2007] Schröder, M. (2007). Interpolating expressions in unit selection. In *proceedings of International Conference on Affective Computing and Intelligent Interaction*.

[Schröder, 2009] Schröder, M. (2009). Expressive speech synthesis: Past, present, and possible futures. *Affective Information Processing*.

[Schroeder, 1999] Schroeder, M. R. (1999). *A Brief History of Speech*. Springer.

[Schroff et al., 2015] Schroff, F., Kalenichenko, D., and Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. *In proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[Shah et al., 2021] Shah, R., Pokora, K., Ezzerg, A., Klimkov, V., Huybrechts, G., Putrycz, B., Korzekwa, D., and Merritt, T. (2021). Non-autoregressive TTS with explicit duration modelling for low-resource highly expressive speech. *In proceedings of International Workshop on Semantic Search Over the Web (SSW)*.

[Shen et al., 2020] Shen, J., Jia, Y., Chrzanowski, M., Zhang, Y., Elias, I., Zen, H., and Wu, Y. (2020). Non-attentive Tacotron: Robust and controllable neural TTS synthesis including unsupervised duration modeling. *ArXiv*. `https://arxiv.org/abs/2010.04301`.

[Shen et al., 2018] Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R., Saurous, R., Agiomyrgiannakis, Y., and Wu, Y. (2018). Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Sini et al., 2018] Sini, A., Lolive, D., Vidal, G., Tahon, M., and Delais-Roussarie, É. (2018). SynPaFlex-corpus: An expressive French audiobooks corpus dedicated to expressive speech synthesis. *proceedings of International Conference on Language Resources and Evaluation (LREC)*.

[Skerry-Ryan et al., 2018] Skerry-Ryan, R. J., Battenberg, E., Xiao, Y., Wang, Y., Stanton, D., Shor, J., Weiss, R. J., Clark, R., and Saurous, R. A. (2018). Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron. In *proceedings of International Conference on Machine Learning, ICML*.

[Snyder et al., 2018] Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-vectors: Robust DNN embeddings for speaker recognition. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Sohl-Dickstein et al., 2015] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics. *In proceedings of Machine Learning Research (MLR)*.

[Sohn, 2016] Sohn, K. (2016). Improved deep metric learning with Multi-class N-pair loss objective. In *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Song and Ermon, 2020] Song, Y. and Ermon, S. (2020). Improved techniques for training score-based generative models. *In proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Song et al., 2021] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. *In proceedings of International Conference on Learning Representations (ICLR)*.

[Sotelo et al., 2017] Sotelo, J. M. R., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A. C., and Bengio, Y. (2017). Char2wav: End-to-end speech synthesis. In *proceedings of International Conference on Learning Representations (ICLR)*.

[Sousa, 2016] Sousa, C. A. R. (2016). An overview on weight initialization methods for feedforward neural networks. *In proceedings of International Joint Conference on Neural Networks (IJCNN)*.

[Ståhl et al., 2005] Ståhl, A., Sundström, P., and Höök, K. (2005). A foundation for emotional expressivity. In *proceedings of conference on Designing for User experience (DUX)*.

[Stan et al., 2013] Stan, A., Watts, O., Mamiya, Y., Giurgiu, M., Clark, R., Yamagishi, J., and King, S. (2013). TUNDRA: a multilingual corpus of found data for TTS research created with light supervision. *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Stanislavski and Vilar, 1984] Stanislavski, K. S. and Vilar, J. (1984). La formation de l'acteur. *Paris*.

[Stevens et al., 1937] Stevens, S. S., Volkmann, J. E., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *Journal of the Acoustical Society of America*.

[Sun et al., 2020a] Sun, G., Zhang, Y., Weiss, R. J., Cao, Y., Zen, H., Rosenberg, A., Ramabhadran, B., and Wu, Y. (2020a). Generating diverse and natural text-to-speech samples using a quantized fine-grained VAE and autoregressive prosody prior. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Sun et al., 2020b] Sun, G., Zhang, Y., Weiss, R. J., Cao, Y., Zen, H., and Wu, Y. (2020b). Fully-hierarchical fine-grained prosody modeling for interpretable speech synthesis. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[T. Bänziger and Scherer, 2012] T. Bänziger, M. M. and Scherer, K. R. (2012). Introducing the Geneva multimodal expression corpus for experimental research on emotion perception. *Emotion*, 12:1161–1179.

[Tachibana et al., 2018] Tachibana, H., Uenoyama, K., and Aihara, S. (2018). Efficiently trainable text-to-speech system based on deep convolutional networks with guided at-

tention. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Takahashi et al., 2006] Takahashi, A., Kurashima, A., and Yoshino, H. (2006). Objective assessment methodology for estimating conversational quality in voip. *IEEE Transactions on Audio, Speech, and Language Processing*, 14:1984–1993.

[Takamichi et al., 2016] Takamichi, S., Toda, T., Black, A. W., Neubig, G., Sakti, S., and Nakamura, S. (2016). Postfilters to modify the modulation spectrum for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24.

[Tan et al., 2021] Tan, X., Qin, T., Soong, F. K., and Liu, T.-Y. (2021). A survey on neural speech synthesis. *ArXiv*. `https://arxiv.org/pdf/2106.15561.pdf`.

[Tao et al., 2014] Tao, J., Hirose, K., Tokuda, K., Black, A. W., and King, S. (2014). Introduction to the issue on statistical parametric speech synthesis. *IEEE Journal of Selected Topics in Signal Processing*, 8:170–172.

[Theis et al., 2016] Theis, L., van den Oord, A., and Bethge, M. (2016). A note on the evaluation of generative models. *In proceedings of International Conference on Learning Representations (ICLR)*.

[Tits et al., 2021] Tits, N., Haddad, K. E., and Dutoit, T. (2021). Analysis and assessment of controllability of an expressive deep learning-based TTS system. *ArXiv*. `https://arxiv.org/pdf/2103.04097.pdf`.

[Toda and Tokuda, 2005] Toda, T. and Tokuda, K. (2005). A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. *IEICE Transactions on Information and Systems*, 90-D.

[Tomczak and Welling, 2016] Tomczak, J. M. and Welling, M. (2016). Improving variational auto-encoders using Householder Flow. *Neural Information Processing Systems Workshop on Bayesian Deep Learning*, abs/1611.09630.

[Toshevska and Gievska, 2021] Toshevska, M. and Gievska, S. (2021). A review of text style transfer using deep learning. *IEEE Transactions on Artificial Intelligence*.

[Tran et al., 2015] Tran, D., Ranganath, R., and Blei, D. M. (2015). Variational Gaussian process. *In proceedings of International Conference on Learning Representations (ICLR)*.

[Tuerk and Robinson, 1993] Tuerk, C. and Robinson, T. (1993). Speech synthesis using artificial neural networks trained on cepstral coefficients. *In proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*.

[Türk et al., 2005] Türk, O., Schröder, M., Bozkurt, B., and Arslan, L. M. (2005). Voice quality interpolation for emotional text-to-speech synthesis. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Usino et al., 2019] Usino, W., Prabuwono, A. S., Allehaibi, K. H. S., Bramantoro, A., A, H., and Amaldi, W. (2019). Document similarity detection using k-means and cosine distance. *International Journal of Advanced Computer Science and Applications*, 10(2).

[Valin and Skoglund, 2019] Valin, J.-M. and Skoglund, J. (2019). LPCNET: Improving neural speech synthesis through linear prediction. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Valle et al., 2020] Valle, R., Li, J., Prenger, R. J., and Catanzaro, B. (2020). Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Valle et al., 2021] Valle, R., Shih, K. J., Prenger, R., and Catanzaro, B. (2021). Flowtron: an autoregressive flow-based generative network for text-to-speech synthesis. *In proceedings of International Conference on Learning Representations (ICLR)*.

[van den Oord et al., 2016] van den Oord, A., Kalchbrenner, N., Espeholt, L., Kavukcuoglu, K., Vinyals, O., and Graves, A. (2016). Conditional image generation with pixelcnn decoders. In *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[van den Oord et al., 2018] van den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., van den Driessche, G., Lockhart, E., Cobo, L. C., Stimberg, F., Casagrande, N., Grewe, D., Noury, S., Dieleman, S., Elsen, E., Kalchbrenner, N., Zen, H., Graves, A., King, H., Walters, T., Belov, D., and Hassabis, D. (2018). Parallel wavenet: Fast high-fidelity speech synthesis. *In proceedings of International Conference on Machine Learning (ICML)*.

[van den Oord et al., 2017] van den Oord, A., Vinyals, O., and Kavukcuoglu, K. (2017). Neural discrete representation learning. In *proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Vasilijevic and Petrinović, 2011] Vasilijevic, A. and Petrinović, D. (2011). Perceptual significance of cepstral distortion measures in digital speech processing. *Journal for Control, Measurement, Electronics, Computing and Communications*, 57:268–281.

[Vásquez-Correa et al., 2018] Vásquez-Correa, J. C., Orozco-Arroyave, J. R., Bocklet, T., and Nöth, E. (2018). Towards an automatic evaluation of the dysarthria level of patients with parkinson's disease. *In proceedings of Journal of communication disorders*.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N. M., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *In proceedings of Conference on Neural Information Processing Systems (NIPS)*.

[Veaux et al., 2017] Veaux, C., Yamagishi, J., and MacDonald, K. (2017). CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit. *University of Edinburgh, The Centre for Speech Technology Research (CSTR)*.

[Vroomen et al., 1993] Vroomen, J., Collier, R., and Mozziconacci, S. J. L. (1993). Duration and intonation in emotional speech. In *proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*.

[Wang et al., 2020] Wang, C., Tang, Y., Ma, X., Wu, A., Okhonko, D., and Pino, J. (2020). Fairseq S2T: Fast speech-to-text modeling with fairseq. *proceedings of the Asia-Pacific Chapter of the Association for Computational Linguistics*.

[Wang et al., 2017a] Wang, J. J., Zhou, F., Wen, S., Liu, X., and Lin, Y. (2017a). Deep metric learning with angular loss. *In proceedings of IEEE International Conference on Computer Vision (ICCV).*

[Wang et al., 1992] Wang, S., Sekey, A., and Gersho, A. (1992). An objective measure for predicting subjective quality of speech coders. *Journal on Selected Areas in Communications*, 14:1890–1901.

[Wang et al., 2017b] Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q. V., Agiomyrgiannakis, Y., Clark, R., and Saurous, R. (2017b). Tacotron: Towards end-to-end speech synthesis. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH).*

[Wang et al., 2018] Wang, Y., Stanton, D., Zhang, Y., Skerry-Ryan, R., Battenberg, E., Shor, J., Xiao, Y., Ren, F., Jia, Y., and Saurous, R. (2018). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *proceedings of International Conference on Machine Learning (ICML).*

[Watts et al., 2016] Watts, O., Henter, G. E., Merritt, T., Wu, Z., and King, S. (2016). From HMMs to DNNs: Where do the improvements come from? *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*

[Watts et al., 2013] Watts, O., Stan, A., Mamiya, Y., Suni, A., Burgos, J. L. M., and Montero, J. M. (2013). Proceedings of Blizzard challenge 2013. *Blizzard Challenge 2013.*

[Weijters and Thole, 1993] Weijters, T. and Thole, J. (1993). Speech synthesis with artificial neural networks. *In proceedings of IEEE International Conference on Neural Networks.*

[Weiss et al., 2021] Weiss, R. J., Skerry-Ryan, R., Battenberg, E., Mariooryad, S., and Kingma, D. P. (2021). Wave-tacotron: Spectrogram-free end-to-end text-to-speech synthesis. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).*

[Weng, 2018] Weng, L. (2018). From autoencoder to beta-VAE . *lilianweng.github.io/lil-log.* `http://lilianweng.github.io/lil-log/2018/08/12/from-autoencoder-to-beta-vae.html`.

[Wensveen et al., 2000] Wensveen, S., Overbeeke, K., and Djajadiningrat, J. P. (2000). Touch me, hit me and i know how you feel: a design approach to emotionally rich interaction. In *proceedings of conference on Designing interactive systems: processes, practices, methods, and techniques (DIS).*

[Wensveen et al., 2002] Wensveen, S., Overbeeke, K., and Djajadiningrat, J. P. (2002). Push me, shove me and i show you how you feel: recognising mood from emotionally rich interaction. In *proceedings of conference on Designing interactive systems: processes, practices, methods, and techniques (DIS).*

[Whiteside, 1998] Whiteside, S. (1998). Simulated emotions: an acoustic study of voice and perturbation measures. *In proceedings of International Conference on Spoken Language Processing (ICSLP).*

[William D Stanley and Saunders, 1988] William D Stanley, Gary R Dougherty, R. D. and Saunders, H. (1988). Digital signal processing. John Wiley and Sons.

[Williams and Stevens, 1972] Williams, C. and Stevens, K. (1972). Emotions and speech: some acoustical correlates. *Journal of the Acoustical Society of America*, 52 4:1238–50.

[Wilson et al., 2021] Wilson, J., Park, S., Wilson, S. J., and Lin, M. C. (2021). Voice aging with audio-visual style transfer. *ArXiv*. `https://arxiv.org/pdf/2110.02411.pdf`.

[Woody, 2000] Woody, R. (2000). Learning expressivity in music performance: An exploratory study. *Research Studies in Music Education*.

[Wu et al., 2015] Wu, Z., Valentini-Botinhao, C., Watts, O., and King, S. (2015). Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Wu et al., 2016] Wu, Z., Watts, O., and King, S. (2016). Merlin: An open source neural network speech synthesis system. *In proceedings of International Workshop on Semantic Search Over the Web (SSW)*.

[Xue et al., 2018] Xue, L., Zhu, X., An, X., and Xie, L. (2018). A comparison of expressive speech synthesis approaches based on neural network. *In proceedings of the Workshop on Affective Social Multimedia Computing*.

[Yamagishi et al., 2004] Yamagishi, J., Masuko, T., and Kobayashi, T. (2004). HMM-based expressive speech synthesis - towards TTS with arbitrary speaking styles and emotions. *proceedings of International Workshop on Innovative Architecture for Future Generation High-Performance Processors and Systems (IWIA)*.

[Yamagishi et al., 2003] Yamagishi, J., Onishi, K., Masuko, T., and Kobayashi, T. (2003). Modeling of various speaking styles and emotions for HMM-based speech synthesis. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Yang et al., 2020a] Yang, F., Yang, S., Wu, Q., Wang, Y., and Xie, L. (2020a). Exploiting deep sentential context for expressive end-to-end speech synthesis. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Yang et al., 2020b] Yang, L., Fan, W., and Bouguila, N. (2020b). Clustering analysis via deep generative models with mixture models. *IEEE transactions on Neural Networks and Learning Systems*.

[Yang et al., 2016] Yang, S., Wu, Z., and Xie, L. (2016). On the training of DNN-based average voice model for speech synthesis. *In proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*.

[Young, 1994] Young, S. J. (1994). The htk hidden markov model toolkit: Design and philosophy. *Technical Report. University of Cambridge: Department of Engineering, Cambridge, UK*.

[Zaïdi et al., 2021] Zaïdi, J., Seute, H., Niekerk, B. V., and Carbonneau, M. (2021). Daft-exprt: Robust prosody transfer across speakers for expressive speech synthesis. *ArXiv*. `https://arxiv.org/pdf/2108.02271.pdf`.

[Zangar et al., 2019] Zangar, I., Mnasri, Z., Colotte, V., and Jouvet, D. (2019). F0 modeling using DNN for arabic parametric speech synthesis. *proceedings of INNS Big Data and Deep Learning (INNSBDDL)*.

[Zen, 2006] Zen, H. (2006). An example of context-dependent label format for HMM-based speech synthesis in English. `http://www.cs.columbia.edu/~ecooper/tts/lab_format.pdf`.

[Zen et al., 2019] Zen, H., Dang, V., Clark, R., Zhang, Y., Weiss, R. J., Jia, Y., Chen, Z., and Wu, Y. (2019). LibriTTS: A corpus derived from librispeech for text-to-speech. *In proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

[Zen et al., 2013] Zen, H., Senior, A., and Schuster, M. (2013). Statistical parametric speech synthesis using deep neural networks. *proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Zen et al., 2009] Zen, H., Tokuda, K., and Black, A. W. (2009). Review: Statistical parametric speech synthesis. *Speech Communication*, 51.

[Zeng et al., 2020] Zeng, Z., Wang, J., Cheng, N., Xia, T., and Xiao, J. (2020). Aligntts: Efficient feed-forward text-to-speech system without explicit alignment. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Zhang et al., 2021] Zhang, M., Zhou, Y., Zhao, L., and Li, H. (2021). Transfer learning from speech synthesis to voice conversion with non-parallel training data. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*.

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

[Zhou et al., 2021] Zhou, K., Sisman, B., Liu, R., and Li, H. (2021). Seen and unseen emotional style transfer for voice conversion with a new emotional speech dataset. *In proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*.

[Zhu et al., 2018a] Zhu, W., Ma, Y. C., Zhou, Y., Benton, M. G., and Romagnoli, J. A. (2018a). Deep learning based soft sensor and its application on a pyrolysis reactor for compositions predictions of gas phase components. In *proceedings of International Symposium on Process Systems Engineering (PSE)*.

[Zhu et al., 2018b] Zhu, Y., Ko, T., Snyder, D., Mak, B., and Povey, D. (2018b). Self-attentive speaker embeddings for text-independent speaker verification. In *proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*.